ICRA-07 Workshop on

Semantic Information in Robotics

April 10, 2007 • Rome, Italy

Proceedings

Editors: Joachim Hertzberg and Alessandro Saffiotti

Program Committee

Michael Beetz (TU München, Germany)
Giuseppe De Giacomo (U. di Roma "La Sapienza", Italy)
Joachim Hertzberg (U. Osnabrück, Germany), Co-Chair
Kurt Konolige (SRI International, Menlo Park, CA, USA)
Ben Kröse (U. Amsterdam, The Netherlands)
Benjamin Kuipers (U. of Texas at Austin, USA)
Alessandro Saffiotti (Örebro U., Sweden), Co-Chair
Gerhard Sagerer (U. Bielefeld, Germany)

CONTENTS

Foreword

There is a growing tendency to introduce high-level semantic information into robotic systems. This tendency is visible in different forms within several areas of robotics. Recent work in mapping and localization tries to extract semantically meaningful structures from sensor data during map building, or to use semantic knowledge in the map building process, or both. A similar trend characterizes the cognitive vision approach to scene understanding. Recent efforts in human-robot interaction try to endow the robot with some understanding of the human meaning of words, gestures and expressions. Ontological information is increasingly being used in distributed systems in order to allow automatic re-configuration in the areas of flexible automation and of ubiquitous robotics. Ontological information was also used recently to improve the inter-operability of robotic components developed for different systems. While all these trends share many common questions and issues, work on each one of them is often pursued in isolation within a specific area, without being aware of the related achievements is other areas.

The **ICRA-2007 Workshop on Semantic Information in Robotics** is part of the 2007 edition of the IEEE International Conference on Robotics ans Automation. The ambition of this workshop is to lay the first stone in building a community of people who are all tackling the problem of using semantic information in robotics, in all its different forms. An important first task for this community is to identify the common questions and concerns, and to start answering them. One such question is: where is the real added value of using semantic information in our field?

This workshop also emphasizes the link between this community and the knowledge representation (KR) community in AI. There are possibly many ideas and formalisms that can be taken from the KR community, but these should be evaluated from the point of view of robotics: any KR formalism of interest needs not only be representationally and inferentially efficient (as normally required for a KR formalism), but also effectively grounded in the robot's sensor and motor signals.

The material presented at the Workshop, and contained in these proceedings, includes six peer-reviewed original contributions and four (non-reviewed) one-page abstracts of poster presentations. The Workshop also features two invited talks by world leading researchers in this field. More information can be found at the workshop home page `http://aass.oru.se/Agora/ICRA07/`.

We hope that you will enjoy this workshop.

Joachim Hertzberg
Alessandro Saffiotti

J. Hertzberg is with the Knowledge Systems Research Group of the Institute of Computer Science, University of Osnabrück, Germany, `hertzberg@informatik.uni-osnabrueck.de`. A. Saffiotti is with the AASS Mobile Robotics Lab of the Department of Technology, Orebro University. `asaffio@aass.oru.se`.

Invited Talks

ICRA-07 Workshop on
Semantic Information in Robotics
April 10, 2007, Italy

Contextualization in Mobile Robots

Daniele Calisi, Alessandro Farinelli, Giorgio Grisetti, Luca Iocchi,
Daniele Nardi, Stefano Pellegrini, Diego Tipaldi, Vittorio Amos Ziparo

***Abstract*— In this paper, we analyze some work on mobile robots with the goal of highlighting the use of contextual information to obtain a flexible and robust performance of the system. In particular, we analyzed the use of context in different robotic tasks, ranging from robot behavior to perception, and then propose to characterize this process of "contextualization" as a design pattern.**

As a result we argue that many different tasks indeed can exploit contextual information and, therefore, a single explicit representation of this information may lead to significant advantages both in the design and in the performance.

I. Introduction

The requirement that robotic systems are flexible and robust to the uncertainties of the environment are becoming more and more compelling, as new applications of robotics in daily life are envisioned. A promising approach to meet this kind of requirements is to organize the system in such a way that some of the processes, that are required on the robot, can be adapted based on information that is not handled by the processes themselves.

Roughly speaking, one could argue that several tasks that are typical of mobile robots can take advantage of information about context. The notion of *context* has been deeply investigated both from cognitive standpoint and from an AI perspective (see for example [12]). In the former case, the study is more focussed on the principles that underlay human uses of contextual information, while in the latter case, the main point is on how to provide a formal account that enables the construction of actual deductive systems that support context representation and contextual reasoning.

In this work, we are interested in discussing to what extent and how the use of contextual information has been advocated in robot design. Therefore, we take a bottom-up approach, which relies on a rough intuitive notion of context, without addressing either a technical definition or a specific representation, and look at specific instances. For example, we are interested in finding systems that can improve the map construction process, by knowing that the robot is currently moving in the corridor of an office building.

More precisely, we are interested in design patterns, where a process is accomplished with general methods, that can be specialized (thus becoming more effective) by taking into account information that is specific to the situation the robot is facing and is acquired and represented "outside" the process itself.

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza" `<lastname>@dis.uniroma1.it`

Dept. of Computer Science, Autonomous Intelligent Systems, University of Freiburg

This design pattern is often regarded as a *hierarchical architecture* [3], where different layers correspond to different levels of abstraction. There are indeed a variety of approaches concerning layered architectures; however, our main concern here is to find interesting instances of the pattern, rather then specific architectural designs.

We call this design pattern *contextualization*, even though it might be confusing with respect to the above cited studies on context.

Consequently, we look at various tasks that are required in mobile robot design and try to provide concrete examples of contextualization. In particular, we first look at contextualization of behaviors, navigation and strategic decisions, such as exploration.We then look at SLAM, where there are already several proposals of contextualization, and other perception tasks.

The result of our analysis is that contextualization can be effectively used in each of the tasks addressed. It seems therefore very appropriate, from an engineering perspective, to build and maintain a single representation of the information that can be contextualized in many different processes.

II. Behaviors

It is broadly agreed that context driven choices are fundamental in robotic scenarios for adapting the behavior to the different situations, which a robot may encounter during execution.

According to Turner [18]:

> A *context* is any identifiable configuration of environmental, mission-related, and agent-related features that has predictive power for behavior.

He proposes a plan selection approach, based on the identification of different contexts, represented as contextual schemas (c-schemas), a frame-like knowledge structure. In c-schemas, slots (or roles) are features of what is being represented, while the filler is a description of the value of the feature. Each c-schema represents a particular context, that is, a particular class of problem-solving situations.

The idea of plan selection is very common, and indeed plan selection is a basic solution to the classical AI planning problem. We are here more focussed on use of context to adapt basic behaviors, since we are concerned with the interface between a symbolic and numerical representation of information. Hierarchical approaches to planning, for example, do not follow our design pattern since the information is always represented in symbolic form.

Typically, basic behaviors require fine tuning of many parameters, which could be adjusted according to contextual information. For example, in a robotic soccer scenario for the 4-Legged RoboCup[1] competition, consider an AIBO robot, which has to grab a ball with its head during a soccer game [7]. The set of parameters controlling the speed (and thus the accuracy) of the behavior depend on whether there are opponents nearby, the ball is near the field sideline and so on. We have experienced that instead of having a proliferation of behaviors for several specific situations, the contextualization of one general behavior based on context is an effective design approach. The characterization of contexts can also be combined with learning techniques and it provides a useful approach to structuring the design of behaviors.

The use of contextual information for behavior specialization is also suggested in Beets & al. [1]. Context is determined using sampling-based inference methods for probabilistic state estimation to deal with noisy and unreliable perceptions. By adopting a probabilistic representation of contextual information it is possible to use it as a parameter of the behaviors, thus allowing for a smooth change of behavior.

III. Search and Exploration

A Search and Exploration task consists of exploring an unknown environment, gathering information about (while building a representation of) it and looking for particular, predefined, features. Contextual information can be relevant in such a high-level task, since as seen in Section II, this kind of information can be used to select the appropriate behavior and plan to reach the goal in a high-level task.

The relevance of the contextual awareness can be seen considering the two most important parts in which a generic search and exploration strategy can be divided: target selection and navigation.

For what concerns the target selection part of the search and exploration strategy, contextual information can be used in several ways, for example to escape a particular area too complex to be explored or where there is a high probability that the robot ends in a situation from where it cannot go out; or else to avoid to explore an area where the probability to find interesting features is considered too low. Since the search and exploration task is a multi-objective task, requiring a choice among, often conflicting, sub-goals (e.g. exploring unknown areas and looking for features in known areas), contextual information can change the relative importance of one kind of sub-goals with respect to the other ones.

Every mobile robot should be provided with the ability to move through the environment in order to make it possible to accomplish its tasks. Although this is a topic that has been studied for decades, there is no general solution for this problem, because the problem is very hard and usually involves many degrees of freedom. For this reason there exist various algorithms and solutions for specific instances of the problem, where some of the constraints can be relaxed or not considered at all. This means also that it is hard to find a single navigation algorithm that can perform well in all the situations that the robot can find during its exploration task. From a high-level point of view, the motion skill can be considered as a robot behavior and the contextual information can help selecting or adapting a particular method or algorithm for this behavior. For instance, the robot can move quickly in areas that are already been explored and searched for features, while it needs to go slowly when it is looking for features (due, for example, to the computational time needed for classification algorithms). Besides of the speed, also the motion algorithm can be (and should be) different and take advantage of the different obstacle configuration and distance from the robot. For example it can be coarse and quick in easy situations and perform a precise motion planning, though computationally heavier, in clutter areas. Moreover the coarse method can try to avoid those situations that can be critical for navigation (e.g. narrow passages, going near obstacles, etc.). However, such situations usually have to be faced when searching for interesting features.

Moreover, the selection of the right navigation method should take into consideration contextual information like openness/clutterness, roughness, if the robot is moving on a skewed plane, etc., and modify the motion algorithm accordingly. Triebel *et al.* [16] use multi-level surface maps to estimate and classify terrain on their traversability level (traversable, non-traversable and wall). Kim *et al.* [8] describe an on-line learning method to predict the traversability properties of complex terrain, exploiting the robot's experience in navigating the environment.

These issues can be extended to multi-robot systems deployed in a search and exploration task. The coordination method may be modified (i.e. it may be tuned accordingly) taking into account environmental information: i.e. the kind of environment the robots are going to explore (e.g. a chemical factory, an office, a collapsed building, etc.). For example, in [17] the coordination algorithm takes into account semantic information related to places in the environment. In particular, the authors show that in an indoor environment long corridors with doors are interesting places to be explored. In fact, sending at least one robot to explore the whole corridor increases the overall performance, because it discovers quickly the structure of the environment and thus it can coordinate better the other robots.

Contextualization in multi-robot coordination is still in a preliminary stage. One main reason, is the difficulty to share a common high-level environment representation among the robots.

IV. SLAM

The problem of Robot Mapping, or the more general problem of Simultaneous Localization and Mapping, is one of the most important and deeply studied aspect of modern robotics. However, this problem has been addressed from a geometrical and numerical point of view, focusing on the underlying estimation process. While good results have been showed, and several working implementations developed, not

[1]www.robocup.org

so many researchers have addressed the problem of how to use semantic and contextual information within the mapping process.

Generally speaking, semantic and contextual information in robot mapping can be classified according to the level of abstraction in the following way (from higher to lower):

- Environment Context,
- Process Context.

In the following, we will explain, for each class, the information it provides and its possible use. It is worth to notice, here, that the first class can be viewed as *human understandable*, in the sense that it can be mainly used in Human Robot Interfaces, as it represents typical knowledge of human being. The second class can be viewed as *robot understandable*, in the sense that it can be mainly used to tune the robot mapping process for efficacy and efficiency purposes. The middle class can be viewed in the middle between those, as it represents information that can be used by other robot processes as well as information that humans can well understand.

A. Environment Context

In high level spatial reasoning, as well as for Human-Robot communication, information regarding the abstract structure of the environment are very useful. Generic concepts, like rooms and corridors, or more specific ones, like Diego's office, can be used by a human user to give easy commands to the robot ("go to Room A" is easier than "go to point 34.5, 42.79).

Up to now, this kind of semantic information is the most studied one. Several approaches extend metric maps with this kind of semantic information. Martinez Mozos *et al.* [11] extracts a semantic topological maps from a metric one using AdaBoost. In a work by Galindo *et al.* [5], a topological map, extracted with fuzzy morphological operators, is augmented by semantic information using anchoring. Diosi *et al.* [4] use an interactive procedure and a watershed segmentation to create a semantic topological map. [9] introduces the paradigm of Human Augmented Mapping, where a human user interacts with the robot by means of natural language processing.

Information about context is mainly used when robotic systems has to be deployed in domestic environment. The Incremental Mapping of [4] or the Human Augmented Mapping of [9] represent valid applications of such information. In our RoboCare project [2], aimed at having an intelligent system to take care of the elderly, some contextual information about the environment have been used. The metric map of the environment has been enriched by semantic labels for the rooms and the objects present there. Those labels were used to monitor the elder during his daily activities and to assist him in tasks like bringing water or food. However, the augmented map was not built autonomously or in a guided tour, but the labeling was done by a human operator. As a future work we will consider techniques to autonomously build this representation.

Another source of semantic information comes from the nature of the environment being mapped. However, this information is mainly used by the human operator, who selects the right algorithm on the mobile platform. No adaptive systems, which detects, for example, indoor or outdoor environments and switch to different SLAM algorithm, have been developed.

B. Process Context

On the lower level, information about the mapping process can be extracted and used. This information could improve the robustness and the speed of current SLAM algorithms. However, not so much works in this direction have been published. To the best of our knowledge, there are only two works that exploits such information explicitly. In [14], two different algorithm are used for incremental mapping and loop closure. Efficient and incremental 3D scan matching is used when mapping open loop situations, while a vision based system detects possible loop closures. The two algorithms are integrated within an EKF filter with a delayed-state map representation. A deeper analysis is carried on our previous work [6]. We discovered three different phases in robot mapping algorithms, namely *exploration, localization* and *loop closure*. We devised an algorithm which is able to detect those phases and tune the computational complexity accordingly. Exploiting this contextual information we were able to drastically speed up classical Rao-Blackwellized mapping algorithm in grid maps.

However, a general framework for using contextual information in the SLAM process has not been developed yet. Even if the mapping phases have been discovered, detection routines are still well-made heuristics, and a good theoretical understanding is not present.

V. Perception

Robot Perception can benefit significantly from contextual information. It is important to notice that when we talk about Perception, we are dealing with sensing modalities. This means not only that it is possible to exploit contextual information to achieve the goal in a specific Perception task, but also that one can use these modalities to achieve information about the context. In Robot Perception, normally, an iterative information process occurs: a top-down analysis, in which the contribution given by the context helps the perception of features and objects in the scene; a bottom-up analysis, in which scene understanding increases the knowledge about the context. This highlights the difficulty in precisely distinguishing contextual information from the information that are directly addressed by the application. We will focus on the contribution given by the context in extracting useful information from the scene, (i.e. the top-down flux), and we will consider only the visual modality.

One task in which the Robot Vision is crucial is navigation. In [10] the objective is to detect the road pixels on an input camera. In this case the contextual information is the knowledge about the direction the car is heading to, that is, straight, left or right. The use of such information is translated in the choice of the opportune template to be used for the road-segmentation task. Here, Robot Vision is also used to retrieve contextual information. What it really matters though, is the design of the system that clearly separates the contribution coming from the main procedure from the contribution given by the contextual information.

Human Robot Interaction offers a variety of applications that make use of contextual information. Even though these kinds of applications are usually conceived with fixed cameras, in some cases the extension to a mobile robot is easy. For instance, in a posture recognition application [15], it is suggested to make use of contextual information such as the kind of environment represented in the scene (i.e, office rather than a gym) to extend the classifier with an a-priori distribution over the postures. More specifically, a Hidden Markov Model is used to filter the state (i.e. posture) transition probability. The values of the (posture) state transition matrix can be tuned taking into account the same kind of contextual information as above, or otherwise considering details specific to the person.

Another interesting case in which integration of the contextual information leads to a higher performance, is the work done by [13]. In this application, an ECA (Embodied Conversational Agent) talks with a user, while it tries to detect his/her head gesture, that is, to detect the nod and the shaking of the head. This recognition is based on the integration of the visual input and the lexical features, the punctuation features and the timing features in the sentence proposed by the ECA. The results clearly show the improvement in the performance of the classification task when using contextual information.

VI. Conclusions

In this paper we have proposed a notion of contextualization, which is based on a simple intuition about the context of operation and captures a design pattern that can be found in several robotics applications. The idea is to make systems more adaptive by exploiting information about the environment or its internal processes in order to improve some of the basic tasks that are typical of mobile robots. In particular, we focussed on behaviors and plans, search and exploration strategies, SLAM and perception in general.

The results of our analysis (not meant to be exhaustive) suggest that there are indeed a few systems, where one can find instances of contextualization. As it turns out, the suggested design pattern is often applicable, while it is not so common to find, because the information extracted from context are often not represented explicitly, but hard coded in the specific techniques.

It seems natural, at this stage, to build a single representation of this kind of information, pulling it out from various system components. A uniform and shared representation of context can lead to two types of advantages: first an improvement in the acquisition and management of contextual information; second, an increased ability of the system to analyze its internal status and recover from malfunctioning that often block the robot operation in the face of unexpected circumstances.

References

[1] M. Beetz, T. Arbuckle, M. Bennewitz, W. Burgard, A. Cremers, D. Fox, H. Grosskreutz, D. Hahnel, and D. Schulz. Integrated plan-based control of autonomous service robots in human environments, 2001.

[2] Amedeo Cesta, Gabriella Cortellessa, Maria Vittoria Giuliani, Luca Iocchi, Riccardo G. Leone, Daniele Nardi, Federico Pecora, Riccardo Rasconi, Massimiliano Scopelliti, and Lorenza Tiberio. The robocare assistive home robot. *Autonomous Robots*, 2007. to appear.

[3] Ève Coste-Manière and Reid G. Simmons. Architecture, the backbone of robotic systems. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 67–72, San Francisco, CA, USA, 2000. ISBN: 0-7803-5889-9.

[4] A. Diosi, G. Taylor, and L. Kleeman. Interactive slam using laser and advanced sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1103– 1108, Barcelona, Spain, 2005.

[5] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3492–3497, Edmonton, CA, 2005. Online at http://www.aass.oru.se/˜asaffio/.

[6] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi. Speeding up rao blackwellized slam. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 442–447, Orlando, FL, USA, 2006.

[7] L. Iocchi, D. Nardi, L. Marchetti, and V. A. Ziparo. S.P.Q.R. Legged Team Description Paper, 2006.

[8] Dongshin Kim, Jie Sun, Sang Min Oh, James M. Rehg, and Aaron Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Orlando, FL, USA, 2006.

[9] Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. Clarification dialogues in human-augmented mapping. In *Proc. of the 1st Annual Conference on Human-Robot Interaction (HRI'06)*, Salt Lake City, UT, March 2006.

[10] Paolo Lombardi, Bertrand Zavidovique, and Michael Talbert. On the importance of being contextual. *Computer*, 39(12):57–61, 2006.

[11] O. Martínez Mozos and W. Burgard. Supervised learning of topological maps using semantic information extracted from range data. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2772–2777, Beijing, China, 2006.

[12] John McCarthy and Buvač. Formalizing context (expanded notes). In Sasa Buvač and Łucia Iwańska, editors, *Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language*, pages 99–135, Menlo Park, California, 1997. American Association for Artificial Intelligence.

[13] Louis-Philippe Morency, Candace Sidner, Christopher Lee, and Trevor Darrell. Contextual recognition of head gestures. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 18–24. ACM Press, 2005.

[14] P. Newman, D. Cole, and K. Ho. Outdoor slam using visual appearance and laser ranging. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1180–1187, Orlando, FL, USA, 2006.

[15] S. Pellegrini and L. Iocchi. Human posture tracking and classification through stereo vision. In *Proc. of Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, pages 261–269, 2006. ISBN: 972-8865-40-6.

[16] Triebel R., P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[17] C. Stachniss, O. Martínez-Mozos, and W. Burgard. Speeding-up multi-robot exploration by considering semantic place information. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1692–1697, Orlando, FL, USA, 2006.

[18] Roy M. Turner. Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies*, 48(3):307–330, 1998.

High-level Interpretation of the Visual Environment

Bernd Neumann

In the last five years interest in artificial cognitive systems with high-level competences such as situation understanding, intention recognition, autonomous learning and common-sense reasoning, has increased considerably. High-level interpretation of the visual environment constitutes an important basis for these cognitive capabilities. In this talk it is shown that the interpretation of visual data can be realized within a formal knowledge-representation framework. An interpretation typically amounts to "explaining" visual data as part of a larger whole. For example, a driver assistant observing street traffic may interpret a pedestrian motion towards the curb as part of a plan to cross the street (and may react accordingly). A knowledge base supporting high-level interpretations must therefore provide a conceptual basis for part-whole reasoning in terms of "aggregates" which specify which roles parts play in a larger context. Formally, part-whole reasoning can be modelled as abduction, i.e. as constructing an explanation for visual data based on conceptual knowledge about aggregates. This suggests that high-level interpretation of visual data can be implemented by standardized procedures within a formal knowledge representation framework. In the talk, the contours of such procedures are sketched for knowledge representation using Description Logics, and some open problems are pointed out. It is shown that a preference measure is required for the selection between logically equivalent interpretations. A probabilistic model of the aggregate hierarchy allows efficient preference computations if the hierarchy fullfills certain abstraction properties.

B. Neumann is with the Institute of Computer Science, University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany, neumann@informatik.uni-hamburg.de

Technical Papers

Towards a Cognitive Architecture for Mobile Robots in Intelligent Buildings

Francesco Capezio, Fulvio Mastrogiovanni,
Antonio Sgorbissa and Renato Zaccaria

***Abstract*— This paper presents a distributed hybrid architecture for knowledge representation and data fusion for Robotics and Ambient Intelligence applications. The work is motivated by the need to adopt a common framework to deal with different aspects of a "smart space". The overall architecture is based on the idea that an intelligent space can be thought of as an ecosystem composed by cooperating artificial entities. These entities collaborate with each other to perform an intelligent multi-sensor data fusion according to the guidance of an active classification layer. Next, this information is used to guide the (possibly coordinated) behavior of mobile robots and intelligent appliances, thus extending the system capabilities. The approach has been thoroughly tested in simulation, and part of the architecture has been exploited in many applications.**

I. Introduction

Ambient Intelligence (AmI) is aimed at designing systems which are able to acquire information from the environment in order to build coherent context models to be used in decision making. These models are built by collecting sensor data through networks of devices; their use involves reasoning about inferred predictions and establishing well-defined interface models with users. The final result is an actual interaction with end users on the basis of functional services (e.g., providing them with directions in unknown environments, aiding visually or mobility impaired people, etc.).

During the past few years, several approaches have been proposed to support the idea of a *smart space*, i.e., a functional interface to services designed to improve the users' quality of life. A smart space requires efficient methods to manage the information flow exchanged by such services. The complexity depends on the interaction among several disciplines: health care monitoring [1], mobile robotics [2][19], intelligent surveillance applications [3][4], knowledge representation [5], state estimation [20] etc.

The exponential growth of the networks connecting devices to cognitive systems requires the evaluation and aggregation of possibly ambiguous information. Therefore, reliable knowledge representation and data fusion techniques must be used in order to provide the architectural infrastructure with an unifying framework. The goal of data fusion is to *maximize* the useful information content acquired by heterogeneous sources in order to infer *relevant* situations and events. Data fusion is not an isolated process: on the contrary, it must be supported by *a priori* knowledge modeling data interpretation, and it must cooperate with systems providing data acquisition and segmentation, filtering, etc.

Several data fusion models have been designed and successfully used in different applications reported in literature. In particular, the extended JDL (Joint Directors of Laboratories) model [6] and its improvements [15][16] hypothesize that information acquisition can be modeled as a 5-layer process. With respect to this framework, current Robotics and AmI system architectures mainly address data fusion of *numerical* or *sub-symbolic* information. Beside the implementation of the individual tasks performed within an intelligent space, data fusion is usually managed using techniques derived from the Bayesian framework, such as Kalman Filters [7] or Relational Markov Networks [20]. A similar approach, but in a decentralized fashion, is undertaken in [21], where an "active sensor network" is exploited for distributed information gathering. Sub-symbolic techniques, such as Fuzzy Logic, are used in [4][8]. As regards the integration between mobile robots and intelligent environments, preliminary attempts can be found in [2][9][19].

In this paper we propose a distributed cognitive architecture able to actively select the sources of information needed to further classify current situations, according to some predefined model formally represented within the system, and to successfully coordinate the behavior of mobile robots and intelligent buildings. In our case, we focus on the design of an indoor pervasive system supervising user activities and events, able to react to emergencies whenever they arise.

We suggest that, with respect to autonomy, robustness and capability issues, the overall system is definitely improved by the introduction of a knowledge representation subsystem. In Section II we introduce the main concepts related to our architectural approach. In Section III, we focus on its distributed data fusion capabilities in the context of highly integrated Robotics and AmI applications. In Section IV we describe the representation at the symbolic level, detailing how this layer guides the low level knowledge acquisition process. Next, actual implementation and experimental results are presented and discussed. Conclusion follows.

II. An Ecosystem of Artificial Entities

According to [10], an ecosystem is defined as "an area of nature that includes living organisms and non-living substances that interact to produce an exchange of materials between the living and non-living parts". In AmI this

F. Capezio, F. Mastrogiovanni, A. Sgorbissa and R. Zaccaria are with DIST, Department of Communication, Computer and System Sciences, University of Genova, Via Opera Pia 13, 16145, Genova, Italy. Email: {francesco, fulvio, sgorbiss, renato}@dist.unige.it

definition can be extended to support the intuition that a smart space can be considered an ecosystem whose artificial entities exchange information for the fulfillment of some common goal. In our architecture we model the system behavior as a decentralized process which is managed by several cognitive-oriented entities. Each entity independently and asincronously processes simple *pieces* of information, thus mapping the system onto the ecosystem structure.

In our vision of an Artificial Ecosystem (henceforth referred to as Æ), the base building block is the concept of "agent": the Æ is a set of m agents attending different tasks. An agent α is a 4-element vector, $\alpha = \langle \theta, \gamma, \iota, \omega \rangle$, where θ specifies the agent capabilities, γ is a set of goals that α can contribute to achieve, and ι and ω are operators returning, respectively, the set of data needed by α to perform its tasks and the set of data produced by α. Being part of an ecosystem, agents can aggregate in more complex *niches*, characterized by common goals g to fulfill. We define a goal g as the result of n cooperating agents. In particular, $g = \omega(A_g)$, where $A_g = \{\alpha_j : j = 1, ..., n\}$ is the niche of the involved agents, and ω is the previously introduced operator returning the *relevant* output data of A_g. With respect to a higher level of complexity, niches themselves can be modeled as agents and, as such, they are possibly part of larger niches. Therefore we extend our definition of "agent" adopting a recursive definition such that $\alpha_k = A_g = \{\alpha_j : j = 1, ..., k-1, k+1, ..., n\}$.

Agents within an Æ can be assigned different roles. Some of them (which we could describe as *device*-related agents $\{\alpha^d\}$) are responsible for managing physical devices such as sensors for data acquisition or actuators; they are characterized by limited sensing or actuating capabilities, and by a low processing power; thus, they can be programmed to perform simple data analysis and react to anomalous data by inspection (e.g., checking if a numerical value is over a given threshold). Other agents (possibly referred to as *cognitive*-related agents $\{\alpha^c\}$) are designed to perform cognitive data interpretation in order to guide the system behavior: in particular, they can concurrently achieve data filtering, feature extraction, symbolic inferences, knowledge representation, data fusion and planning; because they process complex information, they require sophisticate capabilities in processing and exchanging data.

We assume that $\{\alpha^d\}$ agents are *embedded* within the devices they are dealing with (thus implementing a sort of distributed control), while no assumption is made about embodiment of $\{\alpha^c\}$ agents. Therefore, even cognitive agents managing mobile robots-related tasks (e.g., obstacle avoidance, map building, etc.) could be scheduled on fixed workstations instead of being executed on-board. For this reason, we assume that, for each pair (i, j) of communicating agents, a communication *channel* $c_{i,j}$ can be established. Formally, $\forall \alpha_i, \alpha_j \exists c_{i,j}(\alpha_i, \alpha_j)$, where α_i and α_j are cooperating agents such that (a subset of) $\iota(\alpha_i)$ is provided by (a subset of) $\omega(\alpha_j)$ or *viceversa*.

For convenience, $\{\alpha\}$ can be grouped in particular niches which are functionally autonomous. In our Æ we identify three main types of niches: Intelligent Buildings $Æ_{ib}$, Mobile Robots $Æ_{mr}$ and Knowledge Managers $Æ_{km}$. Intelligent buildings and mobile robots are essentially software frameworks (built on top of specific hardware configurations) able to perform sub-object and object assessment. Knowledge managers are software architectures whose aim is to formally represent situations and to infer events which are useful for the system, planning a course of action to issue high-level commands to intelligent buildings and mobile robots. In other words, knowledge managers implement the last 3 layers (situation and impact assessment, process refinement) of a typical JDL model [6].

III. Distributed Configurations of Agents

During the past few years, research in software architectures for smart spaces and mobile robots mainly focused on numerical and sub-symbolic techniques for human-robot interaction, haptic interfaces, state estimation, etc. Within the general framework of the JDL extended model, only the first 2 layers were thus investigated in detail, i.e., sub-object and object assessment. $Æ_{ib}$ and $Æ_{mr}$ are not an exception: they are groups of agents dealing with numerical data acquisition, filtering and segmentation, state estimation, low-level man-machine interfaces, etc. On the contrary, $Æ_{km}$ faces explictly the problem of symbol grounding [11], knowledge representation, and planning-mediated decision making. In this Section, we describe the functional structure of these subsystems.

A. Intelligent Buildings

An $Æ_{ib}$ niche deals with the physical space provided with intelligent devices. It consists both of $\{\alpha^d\}$ and $\{\alpha^c\}$ agents. The first group controls sensors like cameras, PIR (Passive Infra Red), smoke and temperature detectors, or actuators like controllers for automated doors or windows; the second one implements several cognitive behaviors, e.g., acquisition, filtering and processing of raw data in order to extract feature based information, implementation of control system laws for references values to be issued back to $\{\alpha^d\}$, etc. These capabilities are achieved through a tight coupling between device and cognitive agents: suitable groups of agents can be designed in order to arrange incoming data into well-defined formats.

Let's illustrate this process using the following example. Consider a distributed user tracking system based on camera images. This task could involve the fulfillment of the goal g_{bb}, i.e., to obtain bounding boxes from image data. This can be achieved by instantiating the following agent $\alpha_{bbe} = A_{bbe} = \{\alpha^d_{cd}, \alpha^c_{be}\}$, where *bbe* stands for "bounding boxes extractor", α^d_{cd} is a camera device and α^c_{be} is an agent implementing a bounding box extraction algorithm.

In our actual set-up, we dedicate one cognitive agent to manage the behavior of several homogeneous device agents. For example, α^c_{smoke} is a cognitive agent responsible for tracking the current status of all the smoke detector devices $\{\alpha^d_{smoke}\}$ in the system; upon installation of a new α^d_{smoke} network node, the device agent notifies α^c_{smoke} about itself;

at each time step, α^c_{smoke} maintains a numerical representation about the status of all the network of smoke detectors. Analogous considerations hold for different sensors. This approach improve the system tolerance to emergency situations: while each single α^d_{smoke} can immediately fire an alarm on a sensor basis, α^c_{smoke} can reason about the topology of the alarms, being able, e.g., to provide a comprehensive map of the unsafe areas.

Other cognitive agents are responsible for more sophisticate sub-symbolic reasoning schemata. In particular, specific groups of agents have been implemented for user tracking using probabilistic state estimation techniques, ultimate presentation for user interfaces or interactions with other architectural modules, such as $Æ_{mr}$ or $Æ_{km}$: these agents are able to receive high level goals issued by a $Æ_{km}$, or to interact with a $Æ_{mr}$ in order to fulfill complex goals in cooperation.

B. Mobile Robots

A $Æ_{mr}$ is constituted by several groups of agents dealing with the complex behavior of mobile robots. In our approach, mobile robots are situated within an intelligent building, thus being surrounded by several intelligent devices. Despite their claimed autonomy within such environments, they can be considered a mobile *extension* to the $Æ_{ib}$, cooperating with it in order to carry out complex service tasks, e.g., objects delivery and retrieval, surveillance, etc.

Analogously to $Æ_{ib}$, complex behaviors are achieved through cooperation among device- and cognitive-oriented agents. For example, consider the self-localization of a mobile robot with respect to an *a priori* map. The robot is equipped with a laser rangefinder, providing range measurements at discrete interval times. This task could require to achieve the goal g_l, namely, to extract line-based features from the raw scan points. It is possible to define an agent $\alpha_{le} = A_{le} = \{\alpha^d_{ld}, \alpha^c_{le}\}$, where le stands for "lines extractor", α^d_{ld} is a laser driver and α^c_{le} implements a typical line extraction technique [17].

Specifically, on-board $\{\alpha^d\}$ agents manage raw data dealing with sensors used for localization, navigation, obstacle avoidance (e.g., laser rangefinders or sonars), etc., or actuators like motors for locomotion or communication boards. Moreover, they interact with $\{\alpha^c\}$ for the robot to exhibit pure reactive behaviors. Beside a base layer for "low level" robot management (e.g., agent real time scheduling or configuration), "high level" $\{\alpha^c\}$ agents are organized in functional subgroups dealing with well-specified tasks. Groups have been developed for localization and map building (comprising, e.g., Kalman Filters α^c_{kf} and motion model agents α^c_{mm} to compute the robot pose $s = (x, y, \theta)^T$ in a cartesian reference frame), navigation, path planning, etc. [9]. Within each group, cognitive agents manage such diverse tasks as data filtering and segmentation, implementing algorithms for data fusion and interpretation, etc.

Specific software interfaces exchange data with and receive commands from $Æ_{km}$, and cooperate with $Æ_{ib}$ to purposively interact with the environment: e.g., during a mission specified by $Æ_{km}$, mobile robot subsystems can communicate with $Æ_{ib}$ through a particular physical agent, α^p_{bi}, where bi stands for "beacon interface"; the communication is aimed at implementing self-localization through triangulation techniques, or at requesting particular actions to be performed by $Æ_{ib}$ to help robot navigation (to open automated doors, to call an elevator for floor switching, etc.).

C. Knowledge Managers

The main goal of $Æ_{km}$ is to arrange numerical data acquired by $Æ_{ib}$ and $Æ_{mr}$ in models relating information to symbols, i.e., the symbol grounding [11]. Still unsolved in its general formulation, the symbol grounding problem arises whenever symbolic capabilities are introduced into artificial systems.

$Æ_{km}$ deals with symbolic knowledge representation and data fusion by introducing a new class of agents $\{\alpha^c_{kb}\}$, managing knowledge bases represented using Description Logics (DLs). DLs consist of a *Terminology Box* (TBox), modeling concepts, descriptions and relationships (namely, roles among concepts), and an *Assertional Box* (ABox), describing the actual scenario using the ontology described by the TBox. $\{\alpha^c_{kb}\}$ agents allow symbolic data fusion and representation using two *layers*: a model of the physical space to interact with and its related *information space*, i.e., "objects" and devices of the physical world, along with their associated information; the data fusion structure, responsible for the creation of meaningful concept instances (henceforth called "situations") from base predicates corresponding to sensor data. In order to update the ABox in real time, we assume the availability of a comprehensive niche of agents providing $\{\alpha^c_{kb}\}$ agents with heterogeneous information, i.e., $Æ_{ib}$ and $Æ_{mr}$.

In our work we adopt a decentralized approach to information representation. Intelligent buildings and mobile robots are to be considered functionally autonomous with respect to knowledge managers; nonetheless, their behavior is heavily improved by the introduction of such a symbolic counterpart. Representation is distributed for several reasons: among them, efficiency and robustness to possible system faults. Efficiency is a major issue in symbolic knowledge representation. Inferences should be perfomed in such a way that the overall system is able to react within predictable periods of time. In DL-based knowledge bases, the computational complexity of inferences grows exponentially with the number of concepts [22]. In order to reduce the number of concepts, we distribute the overall knowledge base to different agents, thus obtaining loosely coupled knowledge bases. Unfortunately, we observe an overhead in communication due to the need of keeping all the knowledge bases mutually coherents. Future work should hopefully find a good trade-off between centralized and distributed representation. Faults are an issue in multi-agent systems whenever communication among agents is not possible: by decoupling niches of agents, we guarantee a certain level of autonomy for the different subsystems.

Finally, deliberative activities (such as, e.g., planning) are achieved by introducing another class of cognitive agents, namely $\{\alpha_p^c\}$, where p stands for "planning". They receive as input a domain description and a problem formulation, and compute a course of actions – if it exists – to be executed (possibly in cooperation) by agents belonging to $Æ_{ib}$ and $Æ_{mr}$.

IV. Knowledge Representation

Symbolic knowledge representation is performed by the concurrent activities of several cognitive agents, operating on symbols (i.e., instances of `Predicate`) originated within knowledge bases. $Æ_{km}$ is organized in different layers: (i) representation of physical and information spaces, (ii) data fusion, (iii) situation modeling and assessment, (iv) planning and execution.

A. Representation of Physical and Information Spaces

`Entity` is a basic concept for `Object`, `User`, `Robot` and `Building`. Each `Entity` can be characterized by a `Position` within the environment and a current `Situation`. TBoxes maintain a topological representation of the environment. A `Building` is divided in `Place`s and then in `Area`s (further specified in `ToiletteArea`, `StoveArea`, etc.). `Area`s contain instances of `Object`, and descriptions about area navigability (i.e., sequences of `Landmark`s associated with the `Area` itself) and localization (e.g., geometric information about the shape of the `Area`).

Beside common objects (e.g., furniture and appliances), a particular class of objects is `Device`. They are described by a `State`, which is regularly updated through information provided by $\{\alpha^d\}$ agents. `Device`s are classified in `Sensor`s and `Actuator`s. The former group is characterized by a *scope* within the environment, modeled as a collection of areas such that $\forall$`scope.Area`; the latter by a controlled `Object`. Modeled sensors are: `Camera`, `PIR`, `Smoke/GasDetector`, `Odometer`, `LaserRangeFinder`, etc. Examples of actuators are: `AutomatedDoor/Window`, `Wheel`, etc.

`User` models users monitored by the system, interacting with the smart space. `Building` and `Robot` are used as abstract representations of $Æ_{ib}$ and $Æ_{mr}$. Each `Entity` can perform `Action`s. These include: `UserAction`s, `BuildingAction`s and `RobotAction`s. The first class include *passive* actions like `ToBeSomewhere`, or *active* actions like `ReadSomething` or `TurnOnStove`, provided that they can be identified, recognized and well-interpreted by the system; the second class takes into account actions that can be executend by $Æ_{ib}$, encompassing deliberative activities as `CloseWindow`, `AlertSecurity`, `RaiseTemperature`, etc.; the third class depends on the actual mobile robot capabilities: in our scenario, mobile robots are provided with specific domains of expertise, such as navigation (e.g., `MoveFromTo`, `UseElevator`), surveillance (e.g, `PatrolArea`, `NotifySecurityStation`), etc. Deliberative actions are mapped into complex sequences of specific actions which can be predefined (and, as such, modeled as Finite State Machines) or planned on-the-fly according to the current `Situation`s.

An `Agent` is modeled according to the definition given in Section II, using concepts like `Capability`, `Goal`, and `Data`. `Data` is a basic concept for all the data types exchanged within the system. Recall the α_{bbe} agent, introduced in Section 3A. Bounding boxes are modeled as a child concept of `Data`, characterized by a 4-element role specifying `2DPoint`s as the extremes of the bounding box). `DeviceAgent` and `CognitiveAgent` are thus introduced. The former uses an additional role specifying its controlled `Device`: for each `Device` (e.g., `Camera`), a corresponding `DeviceAgent` is introduced (e.g., `CameraAgent`).

In this layer, the ABox contains instances of the concepts `Device`, `Entity`, `Area`, `Data`, etc. Sensory data are mapped to specific instances of `Data`, and then in `Predicate`s, thus updating predefined roles of `Device` instances. They are not given a semantic meaning: therefore, this layer does not suffer from the symbol grounding problem, because association between sensor data and symbols is *a priori* designed.

B. Symbolic Data Fusion

Inferences in DLs are achieved through subsumption. Given two concepts, $\mathcal{C}_1$ and $\mathcal{C}_2$, we say that $\mathcal{C}_1$ is subsumed by $\mathcal{C}_2$ (and we write $\mathcal{C}_1 \sqsubseteq \mathcal{C}_2$) if $\mathcal{C}_2$ is more general than or equivalent to $\mathcal{C}_1$. Subsumption operates on concept descriptions. Given a concept $\mathcal{C}$, we denote its description $\mathcal{D}$ by using an operator δ such that $\mathcal{D} = \delta\mathcal{C}$, and its instances $\mathcal{I}$ by an operator ξ such that $\mathcal{I} = \xi\mathcal{C}$. Symbolic data fusion processes are managed by cognitive agents cooperating with α_{kb}^c. Because each cognitive agent is represented within the TBox, data fusion is an *epistemic* operator $\mathcal{K}$ acting upon concepts described by the input and output data of the corresponding α^c when a particular sensor data configuration is updated.

Let's illustrate this with a couple of examples. Consider first user location tracking. This can be achieved by $Æ_{ib}$ adopting probabilistic frameworks [20]. Symbolic information provided by $Æ_{km}$ is used whenever state probabilities are too ambiguous to be reliable. We consider three sensors: one surveillance camera and two PIRs. A niche is arranged as follows. α_{cam}^d, α_{p1}^d and α_{p2}^d are introduced. α_{cam}^d provides raw images to a cognitive agent α_{bbe}^c. It extracts bounding boxes that are passed to another agent, α_{blob}^c, able to compute color blobs from the bounding boxes. These data are used to associate a specific bounding box with an user, whose dress colors are supposed to be known in advance. Data association is not relevant for this example: it could be possible to use intelligent wearable technology and RFID tags to perform the same association. The key idea is that the arising symbol grounding problem is simplified by the redundancy of the sensor network. α_{p1}^d and α_{p2}^d can provide boolean information about the presence of someone in their surrounding, according to the sensor range and position.

Within the Knowledge Base, `cameraDev` $= \xi$`Camera`, while `pirDev1,pirDev2` $= \xi$`PIR`. Together, α_{bbe}^c and α_{blob}^c are aimed at providing information about user identity, i.e., instances of `UserIdData` $\sqsubseteq$ `Data`. `pirDev1` and `pirDev2` are managed by `pirAgent`. Each `User` is described by a role `id` specifying its identity, such that $\forall$`id.UserIdData`, by a `Position` (i.e., a collection of areas), and by a dress color. This data fusion process is managed by α^c_{cul} (see Algorithm 1), which checks subsumption between the `scope` of each `Device`. With `UserIdData` we identify the `User` whose location is to be computed. It is initialized to $\texttt{cameraDev.scope} = \texttt{area1} \sqcap \texttt{area2} \sqcap \texttt{area3}$. If $\texttt{dp1} = \delta\texttt{pirDev1.scope} = \texttt{area2} \sqcap \texttt{area3}$ and $\texttt{dp2} = \delta\texttt{pirDev2.scope} = \texttt{area2}$, we have $\texttt{d} \sqsubseteq \texttt{dp1} \sqsubseteq \texttt{dp2}$. As a consequence, $\texttt{dp2} = \texttt{area2}$ is the new user location.

Algorithm 1 Compute User Location

Require: $\texttt{id} = \xi\texttt{UserIdData}$; $\texttt{p1}, \texttt{p2} = \xi\texttt{PIRData}$
Ensure: user location
for all $\texttt{u}$ such that $\texttt{u} = \xi\texttt{User}$ **do**
  if $\texttt{id} \sqsubseteq \delta\texttt{u.id}$ **then**
    $\texttt{d} = \delta\texttt{cameraDev.scope}$
    for all $\texttt{p}$ such that $\texttt{p} = \xi\texttt{PIR}$ **do**
      if $\texttt{d} \sqsubseteq \delta\texttt{p.scope}$ **then**
        $\texttt{d} = \delta\texttt{p.scope}$
      end if
    end for
  end if
  $\delta\texttt{u.pos} = \texttt{d}$
end for

Now consider the problem of inferring a mobile robot location given its cartesian pose estimation. The robot is equipped with wheel odometers (represented by `Wheel`) and one laser rangefinder (`LaserRangefinder`). In the ABox, corresponding instances of `Agent` are introduced. $\texttt{RobotPose} \sqsubseteq \texttt{Data}$ is provided by two cognitive agents, i.e., a motion model (e.g., α^c_{mm}) and a state estimation technique (e.g., a Kalman Filter α^c_{kf}). This information is fed to α^c_{crl} (see Algorithm 2), which checks what `Area` encloses the current `RobotPose`.

Algorithm 2 Compute Robot Location

Require: $\texttt{rp} = \xi\texttt{RobotPose}$; $A = \{\texttt{Area1, ..., AreaN}\}$
Ensure: robot location
$\texttt{a}, \texttt{rl} = \delta\texttt{Area}$
for all $\texttt{a}$ such that $\texttt{a} \in \{A\}$ **do**
  if $IsInside(\texttt{rp}, \texttt{a})$ **then**
    $\texttt{rl} = \texttt{a}$
  end if
end for

C. Situation Modeling and Assessment

Deliberative activities are carried out on the basis of the current inferred `Situation`s. In this work, situations are collections of `Predicate`s, stating facts about the current state of the environment. A `Situation` is used to map an `Entity` to some `Action` or event; they are aimed at capturing events occurring within the environment, while `Action`s represent instances of actions fired by an `Entity`.

We model `UserSituation`s, `BuildingSituation`s and `RobotSituation`s. The first class models the status of `User`s. It is detailed using intermediate base concepts (e.g., `BeingSomewhere` or `ReadingSomething`), related to the childs of `Action` (e.g., `ToBeSomewhere` or `ReadSomething`). Thus, `UserSituation` can be further specialized: e.g., `BeingSomewhere` subsumes concepts like `BeingNearToilette`, `BeingInBed`, `BeingNearStove`, etc. Intelligent buildings model through `Predicates`s and `Situation`s complex events occurring within the environment, such as `TemperatureTooHigh`, `SmokeInTheRoom`, etc. Mobile robots represent `RobotSituation`s such as `BeingSomewhere`, `AtFloor`, `Patrolling`, etc. Low-level situations can be used to represent in detail specific robot actions: `WaitingForElevator`, `GoingToNextLandmark`, etc. For each `Entity`, the complete current state can thus be inferred by considering the superimposition of the most recent instances of the `Situation` concept relating the `Entity` itself to each intermediate base concept introduced so far.

As long as new `Data` become available at the symbolic level, instances of `Predicate` and `Situation` are updated within the ABox. The hierarchical `Situation` structure proves to be effective in practice: (i) it is easily extensible: new branches can be added by creating new concepts, if the system is able to distinguish among different situations through (a combination of) sensor data; (ii) its creation can be automated through classification learning; (iii) using the subsumption, a system exploiting the tree for managing an *active* monitoring system can be easily implemented.

Basically, for each epistemic operator $\mathcal{K}$ relative to an entity `e` it is possible to derive a new description $\delta\mathcal{K}$ to be added to its situation `s`. For each branching concept of our `Situation` hierarchy, we must implement an epistemic operator $\mathcal{K}$ providing the necessary $\delta\mathcal{K}$.

Algorithm 3 Classify Situations

Require: $\mathcal{K}$
Ensure: classification or alarm
$\texttt{s} = \xi\texttt{Situation}$
if $\mathcal{K}$ is fired **then**
  $\delta\texttt{s} = \delta\texttt{s} \sqcap \delta\mathcal{K}$
  if $\delta\texttt{s} \sqsubseteq \bot$ **then**
    unexpected classification: fire an alarm
  end if
end if

The overall process can be modeled at the *meta*level using an operator $\mathcal{K}_c$, where c stands for "classification", implementing Algoritm 3. If $\delta\texttt{s}$ is inconsistent, an alarm can be fired. In this way, whenever a `Situation` has child concepts, the system can actively fire the proper epistemic operator to produce the description able to push further the classification. $\mathcal{K}_c$ could be further encapsulated in another agent, $\mathcal{K}_{ac}$, performing an *active* classification procedure over the `Situation` concepts: whenever a `Situation` concept has a non-null set of child concepts, and no information is available from the corresponding operator, the system can decide by purpose to query the corresponding cognitive agent or to instantiate an alternate method to obtain the same information, if available.

D. Planning and Execution

$Æ_{km}$ is responsible for predicting the effects of the overall system behavior over the entities (users, intelligent buildings and mobile robots) modeled by the system itself.

Each `Device` is augmented with a description of its behavior modeled as a state machine. The purpose of this representation is twofold: (i) modeling each device as a *fluent*, i.e., an entity whose state changes in time, in order to reason at the temporal level (e.g., recognizing deadlocks, starvation, etc.); (ii) determining expected state values given the actual device state and inputs, thus being able to detect faults and react accordingly.

The overall behavior of intelligent buildings and mobile robots is explicitly represented as well. Whenever a deliberative process occurs, its effects (a sequence of `Action`s to be performed by the ecosystem) are instantiated within the knowledge base before actually being executed. On the basis of previous work [12], we use a 2-layer planning system tighly coupled with the knowledge representation. In particular, we introduce a new cognitive agent α_p^c (or, alternatively, a new operator $\mathcal{K}_p$) to perform planning tasks on demand. Its requirements are basically a domain specification and a problem definition.

1) From Situations to Problems: Up-to-date information about the system status is provided by instances of the `Situation` concept. Since each `Predicate` is updated by incoming sensor data, the affected `Situation`s are continuously updated as well, thus acting as a *current state*. $\{\alpha_{kb}^c\}$ agents are provided with templates of problems to solve through planning: in other words, a basic `Problem` concept is introduced, which is characterized by roles specifying a *current* and a *goal* state. Both of them must be filled with one or more instances of the `Situation` concept. Whilst the former is automatically filled by the system with the current instances of `Situation` whenever a new $p = \xi$`Problem` is introduced within the ABox, the latter can be specified in different ways: by the system itself (as a consequence of previous inferences), by the user (after she has issued some command to the system), etc. Next, we distinguish in more specific `Problem`s: e.g., for a mobile robot, we have `NavigationProblem`s, `ManipulationsProblem`s (if the robot is equipped with a gripping hand), etc. Corresponding current and goal states are classified, respectively, as `NavigationSituation`s or `ManipulationSituation`s.

2) From Problems to Plans: Once a new `Problem` has been instantiated, a cognitive agent $\{\alpha_{pi}^c\}$ (where *pi* stands for "planning interface") is encharged to translate the required planning information in a format compatible with STRIPS-like planners. In particular, for this work we adopt PDDL3. This step requires the translation of all the required action templates (i.e., relevant instances of the `Action` concept) and of all the instances of the relevant `Object`s. When dealing with specific problems (e.g., a `NavigationProblem`), only the corresponding actions are included (i.e., `NavigationAction`s). This information is then sent to α_p^c. A solution, if exists, is in the form of a plan, i.e., a sequence of actions that should be perfomed by various agents to reach the goal state. $\{\alpha_{pi}^c\}$ is again asked to translate back the plan formulation in the DL-based formalism: i.e., one instance of `Plan`, and corresponding instances of `Action`, are introduced in the ABox.

3) From Plans to Execution: At this point, several approaches are equally possible. We adopt a planning scheme in which there are high-level, generic `Action`s that, when executed, fire low-level, specific `Problem`s to be instantiated and recursively solved by α_p^c, thus interleaving planning and execution. This approach proves to be particularly robust with respect to dynamic and non-predictabile changes in the environment, because we are allowed to disregard in the high level plan specific details about the environment, and to delay the acquisition of current information at the time of low-level planning or execution.

V. Implementation and Experimental Results

In this Section, we first describe the actual implementation of the system, detailing which solutions have been adopted and how they have been exploited. Next, we report about extensive simulation results of cooperation between $Æ_{km}$ and $Æ_{ib}$, and about our current work in real, though simplified, scenario. Finally, we discuss about the interaction among $Æ_{km}$, $Æ_{ib}$ and $Æ_{mr}$ for the fullfillment of a complex navigation mission.

A. Implementation

Actually, $\{\alpha^d\}$ agents exploit the Echelon LonWorks Fieldbus for distributed control; this framework allows reliable concurrent programming techniques and communication protocols to be easily implemented. $\{\alpha^c\}$ agents, on the contrary, are implemented using ETHNOS, a distributed framework developed for Robotics and AmI applications [13]. ETHNOS allows the scheduling of both periodic and aperiodic tasks in a transparent way. The knowledge bases are developed embedding in an ETHNOS agent Classic, a Description Logic which guarantees sound and complete subsumption inferences [22]. Planning capabilities required by $Æ_{km}$ are achieved using a PDDL3 compatible planner. In our experiments, we tried several deterministic planners with different capabilities. Currently, we adopted SGPlan5, winner of the deterministic track of the 2006 International Planning Competition [23].

B. Intelligent Buildings

The AmI-related cognitive agents of our framework have been mainly tested in a simulation environment built using the architecture itself [14] (see Fig.1 on the left). Simulated experimental results are carried out by adding specific agents implementing instances of patterns of user activities, and providing sensors with data sets recorded through real sensors (particular data sets have been developed to simulate fire, gas leaks, etc.). Each base pattern (see Algorithm 4)

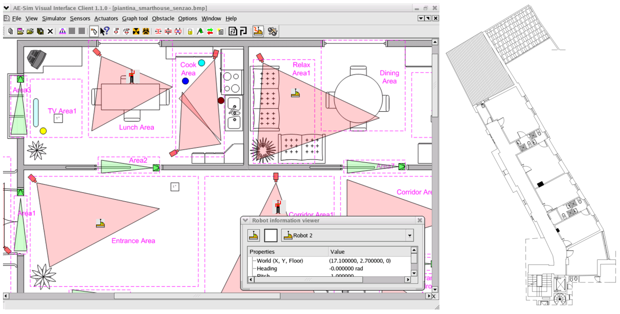


Fig. 1. (Left) The simulation environment; (Right) Map of an on going experimental set-up.

Algorithm 4 Base Pattern

Require: $A = \{a_1, ..., a_n\}$; $E = \{e_1, ..., e_m\}$; E_{pdf}
Ensure: A specific user behavior
for all i such that $i = 1, ..., n$ **do**
  perform action a_i
  choose j according to E_{pdf}
  fire the event e_j
end for

is a sequence of parameterized actions, e.g., movements, interactions with the environment, etc. Examples of base patterns are $A_{answer-phone-call}$ = {answer, talk, hang-up} or A_{lunch} = {goto-kitchen, goto-stove, cook, wait(1), goto-table, eat}. It is worth noting that not all the actions are treated as discrete events: e.g., user movements (i.e., goto-*someplace*) correspond to trajectories in the Cartesian space. Moreover, during each iteration in Algorithm 4, an event e_j is chosen to possibly introduce a perturbation in the current action. Events can range from waiting a certain amount of time to interacting with appliances, from receiving phone calls to completely changing the current pattern with a new one. Events are selected according to a non-uniform outcome probability distribution E_{pdf}. In all the experiments, α^c_{cua} and α^c_{bbe} are able to track the user with cameras, PIRs, lasers and other sensors, maintaining also multiple hypotheses through subsumption. Another agent, α^c_{fa}, is able to fire alarms whenever the user remains in the `BedArea` for more than a specified time, i.e., the corresponding situation `BeingInBed` lasts for too long.

Specific experiments are aimed at testing the active classification system. For example, during the execution of the pattern A_{lunch}, after the user has moved to the `StoveArea`, the event `AnswerPhoneCall` is selected. This implies that the current pattern is replaced by $A_{answer-phone-calls}$. After some time, α^c_{kb} is not updated with the expected information. Thus, a specific query to α^c_{ss} (a cognitive agent managing information coming from a smoke sensor) is made. If the smoke sensor is responding, the system reminds the user about his previous cooking action. On the contrary, if no smoke is detected, it is inferred that the user was doing something else (the user could be asked about it, being questions posed by the system a suitable way to obtain information). After some time, if a `Cook` action is performed, the epistemic operator $\mathcal{K}_{ss}$ operates on the knowledge base in order to infer that the new user situation is `Cooking`; if not, a new classification is made or an alarm is fired.

C. Mobile Robots and Coordinated Behaviors

The overall capabilities of our self-designed mobile platform Staffetta have been previously presented in [9] [12]. In particular, here we report about some recent attainment in cooperation among $\AE_{ib}$ and $\AE_{mr}$ agents.

To understand what happens in practice, let's consider the following scenario, which describes a permanet set-up in our department. The Secretariat Office receives a package, to be forwarded to our laboratory. Through a web interface, which is managed by the α^c_{web} agent in $\AE_{mr}$, a new `TransportationProblem` is instantiated within α^c_{kb}, i.e., to go to the Secretariat Office and bring the package to the Laboratorium. In cooperation with α^c_p, a new high level `Plan` is created as follows: 1) `GoTo(SecretariatOffice)`; 2) `LoadPackage`; 3) `GoTo(Laboratorium)`. In our actual implementation, since Staffetta is not provided with a gripping hand, it is not able to autonomously load a package: therefore, such an `Action` is implemented by simply asking someone to do it for the robot. Let's detail, e.g., the first high level `Action`. It is likely that the actual robot location (which is continuously computed by α^c_{crl}, see Algorithm 2) is different from `SecretariatOffice`; thus, the execution of `GoTo(SecretariatOffice)` involves a complex navigation, possibly with floor switching. Assume that Staffetta is at the second floor, inside the Laboratorium. The execution of the first high level `Action` fires a new `NavigationProblem`: its precondition list is `atFloor(2)` $\sqcap$ `in(Laboratorium)`, while the goal state is `atFloor(0)` $\sqcap$ `in(SecretariatOffice)`. While the precondition list is determined by the actual Staffetta' `Situation`s, the goal state is automatically derived by the high level `Action`. Again, the problem specification is submitted to α^c_p, which produces the corresponding `NavigationPlan`, constituted by a sequence of `MoveFromTo` to reach the elevator, followed by an `UseElevator` action, and then by another sequence of `MoveFromTo` to reach the `SecretariatOffice`.

`MoveFromTo` actions are executed by retrieving in the knowledge base the `Landmark`s (actually, instances of `2DPoint`) that the mobile robot should visit in sequence. This list is fed to the navigation subsystem of $\AE_{mr}$. Once all the landmarks are visited, the single `MoveFromTo` action is accomplished, and $\AE_{km}$ continues the execution.

On the opposite, `UseElevator` requires a tight cooperation with $\AE_{ib}$ (see Fig. 2 on the left). At each floor, near the elevator door, our intelligent environment is provided with a device whose purpose is to manage mobile robot requests to use the elevator. We call this device "beacon", and we formally denote it as α^d_b. Through the α^c_{bi} agent, $\AE_{mr}$ requests the presence of the elevator at the current floor. The information channel between the two agents is managed by an Infra Red channel, extension of the Echelon Fieldbus. This information is received by α^d_b, which notifies α^c_e (a

Fig. 2. (Left) Staffetta entering the elevator; (Right) User tracking through cameras.

cognitive agent managing the behavior of the elevator) about the request. α_e^c maintains a queue of all the requests, and notifies all the α_b^d agents distributed at different floors about the state of the elevator itself. Once the elevator is at the floor and the doors are opened, Staffetta is ready to reach a `Landmark` located inside the elevator itself. This can be accomplished because each event concerning the status of the elevator is notified by α_e^c to all the α_b^d. Thus, through the α_{bi}^c interface, Staffetta extends its cognitive capabilities *observing* the behavior of the elevator. Once inside, using the same principle, Staffetta is continuously notified about the floor switching. We can model this process by assuming that the robot is provided with a virtual *floor sensor*: when the target floor is reached and the doors are opened, Staffetta can exit from the elevator reaching a predefined `Landmark`. The system is actually being tested in a simplified scenario comprising a couple of rooms (a bedroom and a kitchen, see Fig. 2 on the right) which has been fournished with cameras, PIRs, smoke sensors etc., to infer the actions performed inside the rooms, while Staffetta is able to navigate in all the three floors of our Department. Moreover, the overall system is going to be tested at Istituto Figlie di N.S. della Misericordia, Savona, Italy, an assisted-living facility for elderly and disabled (see Fig.1 on the right).

VI. Conclusion

In this paper we presented a hybrid software architecture dealing with knowledge representation and data fusion for integrated Robotics and AmI applications. Despite its simple architecture, it is able to manage heterogeneous information at different levels, "closing the loop" between sensors and actuators. Moreover, while a smart space can extend its capabilities through mobile robots, from the robot perspective its cognitive capabilites are extended through smart space-mediated distributed sensing. The system is able to process numerical as well as symbolic data. Based on the concept of an ecosystem of artificial entities, the system architecture has been thoroughly tested in simulation, and part of it has been used in a real set-up. While actual work is focused on expanding the data fusion capabilities through learning, i.e., adding new branches to the `Situation` based structure, future work will involve an in depth investigation about the integration between all the subsystems in a real, complex, experimental scenario.

References

[1] T. S. Barger, D. E. Brown and M. Alwan, "Health-Status Monitoring Through Analysis of Behavioral Patterns", *in IEEE Trans. on Systems, Man, and Cybernetics – Part A*, vol. 35(1), January 2005.

[2] M. Broxvall, M. Gritti, A. Saffiotti, B.S. Seo and Y.J. Cho, "PEIS Ecology: Integrating Robots into Smart Environments", *in Proc. of the 2006 Int. Conf. on Rob. and Autom. (ICRA)*, USA, May 2006.

[3] Fan Tian Kong, You-Ping Chen, Jing-Ming Xie and Zu-De Zhou, "Distributed Temperature Control System Based on Multi-sensor Data Fusion", *in Proc. of the 2005 Int. Conf. on Machine Learning and Cybernetics*, China, August 2005.

[4] Shaoua Chen, Hong Bao, Xianyun Zeng and Yimin Yang, "A Fire Detecting Method based on Multi-sensor Data Fusion", *in Proc. of the 2003 Int. Conf. on System, Man and Cybernetics (SMC)*, Washington, DC, October, 2003.

[5] J. C. Augusto and C. D. Nugent, "A New Architecture for Smart Homes Based on ADB and Temporal Reasoning", *in Proc. of 3rd Int. Conf. on Smart Homes and Health Telematics*, Canada, July 2005.

[6] F. E. White, "Managing Data Fusion Systems in Joint and Coalition Warfare", *in Proc. of the 1998 Int. Conf. On Data Fusion (EuroFusion98)*, Great Malvern, UK, October 1998.

[7] A.H.G. Al Dhaher and D. MacKesy, "Multi-sensor Data Fusion Architecture", *in Proc. of the 3rd IEEE Int. Work. On Haptic, Audio and Visual Environments and their Applications (HAVE)*, Ottawa, Canada, October 2004.

[8] H. Hagras, V. Callaghan, M. Colley and C. Graham, "A Hierarchical Fuzzy-genetic Multi-agent Architecture for Intelligent Buildings Online Learning, Adaptation and Control", *in Journal of Information Sciences*, vol. 150 (1–2), March 2003.

[9] A. Sgorbissa and R. Zaccaria, "Robot Staffetta in its Natural Environment", *in Proc. of the 8h Conf. of Int. Autonomous Systems (IAS-8)*, Amsterdam, The Netherlands, March 2004.

[10] E. P. Odum, "Fundamentals of Ecology", W. B. Sanders, USA, 1959.

[11] S. Harnad, "The Symbol Grounding Problem", *in Physica D*, vol. 42, pp. 335 – 346, 1990.

[12] F. Mastrogiovanni, A. Sgorbissa and R. Zaccaria, "A System for Hierarchical Planning in Service Mobile Robotics", *in Proc. of the 8th Conf. of Int. Auton. Systems (IAS-8)*, The Netherlands, March 2004.

[13] M. Piaggio, A. Sgorbissa and R. Zaccaria, Pre-emptive versus Non Pre-emptive Real Time Scheduling in Intelligent Mobile Robotics", *in Journal of Exp. and Theor. Artificial Intelligence*, vol. 12 (2), 2000.

[14] F. Capezio, D. Femia, F. Mastrogiovanni, A. Sgorbissa and R. Zaccaria, "AE-Sim 2.0: a Distributed Simulator for Intelligent Spaces and Robots", *in Proc. of the Fourth Int. Conf. on Smart Homes and Health Telematics (ICOST06)*, Northern Ireland, UK, June 2006.

[15] A. Steinberg and C. Bowman, "Revisions to the JDL data fusion model", *in D. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion*, CRC Press LLC, Boca Raton, FL, 2001.

[16] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E.Waltz and F. White, "Revisions and extensions to the JDL data fusion model II", *in P. Svensson, J. Schubert (Eds.), Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, June 2004.

[17] V.T. Nguyen, A. Martinelli, N. Tomatis and R. Siegwart, "A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics", *in Proc. of the 2005 IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS2005)*, Canada, September 2005.

[18] O. Pettersson, "Execution Monitoring in Robotics: a Survey", *in Robotics and Autonomous Systems*, vol. 53(2), pages 73–88, 2005.

[19] J.H. Kim, Y.D. Kim and K.H. Lee, "The Third Generation of Robotics: Ubiquitous Robot", *in Proceedings of the Second Int. Conf. on Autonomous Robots and Agents*, New Zealand, 2004.

[20] L. Liao, D. Fox and H. Kautz, "Location-based Activity Recognition using Relational Markov Networks", *in Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, Edimburg, Scotland, 2005.

[21] A. Makarenko, H. Durrant-Whyte, "Decentralized Bayesian Algorithms for Active Sensor Networks", *in Information Fusion*, vol. 7, pages 418–433, 2005.

[22] H.J. Levesque and R.J. Branchman, "Expressiveness and Tractability in Knowledge Representation and Reasoning", *in Computational Intelligence*, vol. 3(2), pages 78–93, 1987.

[23] W. Hsu, B. W. Wah, R. Huang and Y. X. Chen, "New Features in SGPlan for Handling Soft Constraints and Goal Preferences in PDDL3.0", *in Proc. of the Fifth International Planning Competition, International Conf. on Automated Planning and Scheduling (ICAPS)*, Cumbria, UK, June 2006.

Knowledge Representation in CiceRobot: a Robot for Explorations of Cultural Heritage

Antonio Chella, Marilia Liotta and Irene Macaluso

Abstract— **The aim of the research is to integrate perception, action and symbolic knowledge in order to allow an autonomous robot to operate in unstructured environments and to interact with not expert users. In order to achieve such goals we proposed a cognitive robot architecture based on the integration between subsymbolic and linguistic computations through the introduction of an intermediate level of representation based on conceptual spaces. The architecture has been tested in the CiceRobot project on tasks related to guided tours in the Archaeological Museum of Agrigento. Experimental results show that robot cognitive behaviors allow to achieve a full functional robotic museum guide. In particular, through the interaction with visitors the robot is able to customize the tour depending their preferences. The paper presents a significant case study because it involves perception, planning and human-robot interaction. The proposed architecture addresses the capacities which are generally addressed by an intelligent agent: the capability of representing itself and the external world, of imagining possible evolutions of the world, of paying attention to the relevant events, of planning and evaluating situations and actions.**

I. INTRODUCTION

An autonomous robot operating in real and unstructured environments has to be able to interact with a dynamic world populated with objects, people, and in general, other agents. In order to achieve such goals a robot should be aware of its external and inner perceptions, should be able to pay attention to relevant entities in its environment, to image possible evolutions of the world, to plan its actions and to evaluate situations and plans. Since robots work together people, they would be able to interact with them and to process information coming from this interaction. In the course of the years, the Robotics Lab of University of Palermo developed a robotic architecture that takes into account several suggestions from cognitive science [1] [2]. The architecture is currently experimented on a robot platform based on a RWI-B21 robot equipped with a pan-tilt stereo head, laser rangefinder and sonar (Fig.1). The aim of the architecture is to integrate perception, action and symbolic knowledge representation by means of an intermediate level of representation based on conceptual spaces [3] which provide linguistic symbols with the correct semantic. Moreover, an especial attention on human-robot interaction has been paid in order to allow the system to understand the meaning of what the user asks for and also what it does not explicitly express. Several methodologies have been proposed to take into account semantic content of terms through statistical learning algorithms ([4], [5], [6], [7], [8]). The architecture has been tested in the CiceRobot project on tasks related to guided tours in the Archaeological Museum of Agrigento (see Fig. 1). The proposed architecture allows to deal with the issues related to perception, planning and human-robot interaction typical of museum tour applications. Compared to other related works (see e.g. [9]), our approach mainly focuses on cognitive behaviors which are, in our opinion, fundamental to achieve a full functional robotic museum guide. The museum is arranged both chronologically and topographically, but the sequence of findings to be visited can be rearranged depending on user queries, making a sort of dynamic virtual labyrinth with various itineraries. Therefore, CiceRobot is able to guide visitors both in a prearranged tour and in an interactive tour, built in itinere depending on the interaction with the visitor: the robot is able to rebuild the virtual connection between findings and the path to be followed.

The paper is organized as follows. Sect. 2 describes in details the cognitive architecture; Sect. 3 concerns the robot knowledge representation; Sect. 4 deals with tour building issues; Finally, Sect. 5 is a detailed description of an example of the operations of the robot at work.

Fig. 1. CiceRobot at work.

A. Chella is with the Department of Computer Engineering, University of Palermo, 90128 viale delle Scienze, Italy chella@unipa.it
M. Liotta is with the Department of Computer Engineering, University of Palermo, 90128 viale delle Scienze, Italy liotta@csai.unipa.it
I. Macaluso is with the Department of Computer Engineering, University of Palermo, 90128 viale delle Scienze, Italy macaluso@csai.unipa.it

II. CICEROBOT ARCHITECTURE

The robot cognitive architecture [1] [2] is organized in three computational areas (Fig.3). The Subconceptual Area processes both robot sensors data, allowing to react in real time to unpredictable situations, and user queries, in order to effectively interact with the user. In the Linguistic Area, representation and processing are based on a logic-oriented formalism. The Conceptual Area is intermediate between the Subconceptual and the Linguistic Areas. This area is based on the notion of Conceptual space [3], a metric space whose dimensions are related to the quantities processed in the Subconceptual area. We call knoxel a point in the Conceptual space. Gärdenfors highlights that linguistic and conceptual representation are "different perspectives on how information is described": in this perspective, symbols at linguistic level have been anchored into geometric entities in the conceptual area as in [10].

Knowledge representation is shared between the three areas (see Fig. 2). On the base of the a priori knowledge, a default visit is built, which can be personalized taking into account user queries. The *Query and Documents Processing Module* processes the generic user query in order to compute the LSA Representation allowing to retrieve semantically relevant exhibits as described in section III.B. The system a priori knowledge is maintained in the *Linguistic Knowledge Representation* and anchored to geometric entities of the *Map Repository* (see section III.D). The *Domains Merging* combines geometric and semantic information in order to set in the conceptual space the knoxels related to findings. This allows the *Tour Building* to plan an ad hoc tour (initial plan) taking into account both the spatial relationship between the exhibits and their semantic relevancy to user request, as described in section IV. The *Simulation and Refinement Module* verifies the applicability of the initial plan and, if necessary, modifies it (ideal plan). The robot will not be generally able to exactly follow the ideal plan because of the presence of unknown moving obstacles and also because of the sensory motor errors. In order to deal with such unexpected situations, the *Reactive Executor* controls the robot at execution time. As many of the exhibition windows are invisible to robot sensors, reactive modules take into account also data coming from the 3D simulator in order to perform a reliable obstacle avoidance. To this aim, during the plan execution images processed by the *Vision System* are compared with the corresponding expected scene generated by the simulator. The outcome of the Comparison module is used to localize the robot in order to update its expectations about the environment. See [11] for a more detailed description of the robot architecture.

III. Robot Knowledge Representation

A. Linguistic Area

The Linguistic Area is based on a rich linguistic representation based on the NeoClassic system, a hybrid formalism based on a Description Logic in the KL-ONE tradition ([12], [13]), constituted by a terminological and an assertional components.

In our system, the terminological component contains the description of significant concepts (e.g. showcase, finding and so on). Fig. 3 shows a fragment of the terminological knowledge base. Intuitively, an *Object* is located in a *Room*. A *Finding* is an *Object* described by an *Artistic Description* and located in an *Exhibit Room*. A *Showcase* is an *Object* containing a *Finding*. An *Obstacle* is an *Object* without any *Artistic Description*.

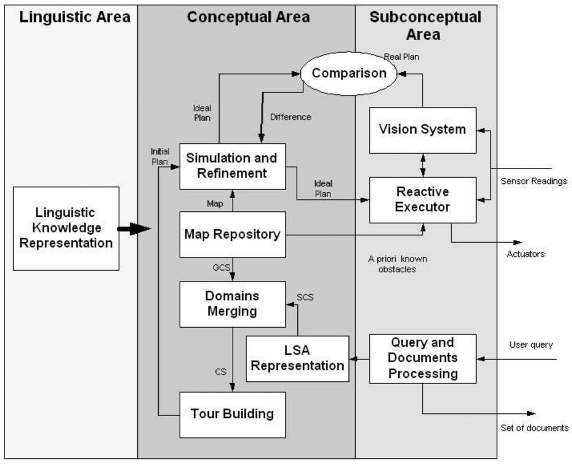


Fig. 2. Architecture

Using a predicative language the assertional component stores facts describing the museum environment. The concepts of the terminological component correspond to one-argument predicates, and the roles (e.g. is_in, is_connected_to) correspond to two argument relations. For example, the existence of an instance *Kore* of the concept *Finding*, is asserted by the formula:

Finding(Kore).

The formula

is_located_in(Kore, Room5)

expresses the fact that the filler of the role *is_located_in* of *Kore* is the term *Room5*.

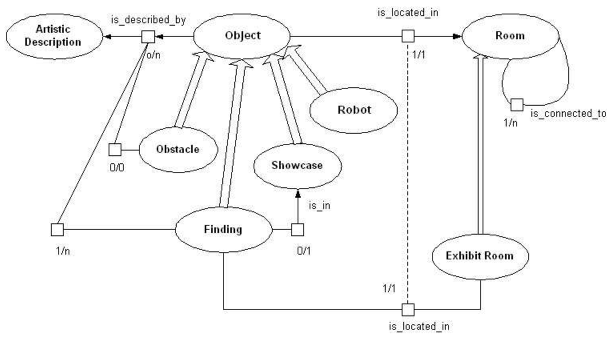


Fig. 3. A fragment of the terminological KB.

B. Conceptual Area

The Conceptual Space (CS) is constituted by two domains: geometric and semantic. A domain is defined as a Conceptual Space having only dimensions that strictly depend each other and are separable from others [3]. We call knoxel a point in the CS, (k_a in Fig. 4). A point $g_a$in the geometric domain (GCS, Geometric Conceptual Space) corresponds to geometric 3D primitives according to the perceived data and the system a priori knowledge. Therefore, the perceived objects, as the robot itself, correspond to suitable sets of points in the robot's GCS. The GCS lets the robot to imagine possible future interactions with the objects in the environment: the interaction between the agent and the environment is represented as a sequence of sets of points imagined and simulated in the Conceptual Area before the execution. A point s_a in the semantic domain (SCS, Semantic Conceptual Space) is a vector coding the semantic distance between the current user query and the each finding. A knoxel in the CS has geometric dimensions g_a and semantic dimensions s_a (see Fig. 4). A configuration of knoxels in CS holds until a new user query is processed. In this case, some knoxels change their positions in CS. The query has the effect of a scattering of the knoxel positions, according to the s_a component. Fig. 4 represents the "scattering" of knoxels corresponding to two different user queries.

The retrieval of findings related to a user query is turned into a semantic document retrieval problem, by associating to each finding a descriptive document. The semantic dimension is computed through algebraic subconceptual processing which leads to a conceptual representation in a vector space where semantic nearness is evaluated by geometric properties. Terms and documents related the museum are represented in this space. A function **f**(.) with following characteristic has to be used:

1) **f**(x) = **v**, **f**() codes the term x in the vector **v**
2) d(**v1**, **v2**)$\longrightarrow$ s(t1, t2) where **v1= f**(t1) and **v2= f**(t2)

d(.,.) is the Euclidean distanced between two vectors; s(.,.) measures the semantic nearness between two terms or documents. To obtain the function **f**(), many theories that try to mine the meaning of words through algebraic methods applied to a large corpus of texts can be used. In the implemented system, the Latent Semantic Analysis (LSA) has been applied. LSA methodology models, in a sub-symbolic way, both the user query and the information to retrieve, terms and documents, as vectors in a vector space. An interesting feature of the space is that terms with similar meaning are coded by near vectors in the space. See [4] for a meaningful example. In the LSA space, an index that measures the semantic distance between two terms, can be determined evaluating geometric distance between the related vectors: terms can be treated independently by their lexical and syntactical representation and information retrieval is based on conceptualization and categorization.

User query is coded in the LSA space [14] through the folding-in technique [15] (see Fig. 5). Therefore, the semantic relevancy of a finding compared to the query is simply evaluated by the angle between the corresponding vectors. In particular the semantic closeness is evaluated as follow:

$sc_i = \cos(\alpha)$,

where α is the angle between the vector coding the current user query and the vector coding the document associated to the i-th finding (see Fig. 5).

For instance, if the query is "*who are chtonic divinities?*", the related *query vector* is represented in the LSA space built as a results of a training phase through a collection of selected documents. For each finding it is evaluated the angle made up by its descriptive document and the query vector : the cosine of this angle is the semantic component. In particular, the finding "*Kore's head*" is represented in the space by a vector near to the query vector: they form an angle having a high value of cosine equal to 0.81, in fact Kore is a chtonic divinity and the finding is strictly semantically related to the query. It means that the semantic value for the "*Kore's head*" in correspondence to the query "*who are chtonic divinities?*" is 0.81.

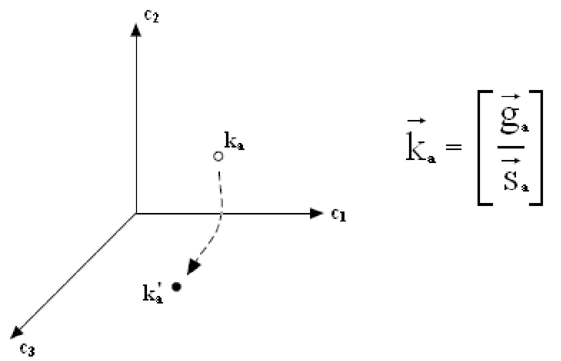


Fig. 4. The black dot corresponds to the knoxel related to the current query configuration, k'_a, the white dot corresponds to the knoxel related to the previous query configuration, k_a.

Moreover, documents strictly related to user query are retrieved both on a local repository and in Internet [11].

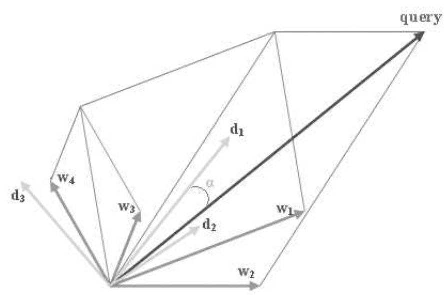


Fig. 5. Vector query representation obtained with the folding-in technique. The query q is made up by terms [w1, w2, w3, w4]; the vector **q** coding the query is obtained as:**q=w1+w2+w3+w4**.

C. SubConceptual Area

Standard subconcetual techniques have been applies to address issues related to document retrieval. They are not treated in this paper. For a deep explanation see [11].

D. Anchoring

Knowledge representation is shared between the three areas; each of them focuses on different aspects of the same entity. The right correspondence between these representation has to be create and maintained; i.e. the anchoring problem should be solved. As shown in [10], Conceptual Space offers a useful intermediate level of representation to anchor symbolic and perceptual data.

The general framework proposed in [10] has been adapted to address the perceptual constraints of the current experimental setup. In particular, as we can assume that objects in the museum are static and as light conditions make very difficult to extract steady features from images, the mapping between symbolic and conceptual representation depends on the a priori known objects position, on the Hue value on the HSV space and on shape parameters (contours). Even considering as static objects and consequently knoxels in CS, a tracker to keep aligned the symbolic and conceptual representation has been introduced in order to deal with uncertainty caused by the robot localization process and small changing in position of findings due to cleaning and maintaining operations.

IV. Tour Building

The CS allows to build an ad hoc tour taking into account both the spatial relationship between the exhibits and their semantic relevancy to user request. A connectivity graph G(V,E) contains the relations between rooms in the museum. Each node v_n corresponds to a room, while an arc e_k between two nodes exists if the corresponding rooms are connected. We denote with $e_k = e_{mn} = (v_m, v_n)$ the oriented arc connecting rooms v_m and v_n. The related cost $c_{mn} = c(v_m, v_n)$ is the Euclidean distance between rooms.

When a query is inserted, the costs are consequently updated according to the following steps:

for all nodes $v_n \in V$ *do*

$s_n = \max_i\{sc_{in}\} + 2$

end for

$c_{mn} = \frac{c^t_{mn}}{s^q_n}$

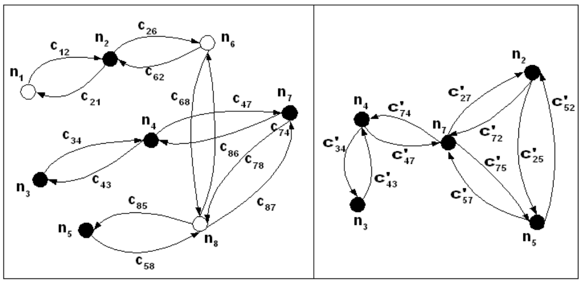
where sc_{in} is the semantic value of the i-th finding in the n-th room; t and q respectively weight the geometric and the semantic components depending on the application context [3]. It should be pointed out that $\forall$ n $s_n \geq 1$. In particular if $s_n = 1$, the semantic value of the related room is null. Instead, if $s_n > 1$ the n-th room contains at least one finding with a not null semantic relation with the query; therefore the cost c_{mn} decreases.

The original set of nodes V is partitioned by sets $B = \{b_1, b_2, ., b_r\}$ and $W = \{w_1, w_2, ., w_s\}$, where b_n is the n-th room with $s_n > 1$ and, w_n is the n-th room with $s_n = 1$ (respectively black and white nodes in Fig. 6). A reduced graph G'(B,E') is computed considering only nodes in B. Denoting with $e'_k = e'_{mn} = (b_m, b_n)$ the oriented arc connecting rooms b_m and b_n, the related cost $c'_{mn} = c'(b_m, b_n)$ is computed as follow:

$$c'_{mn} = \begin{cases} c(b_m, b_n) & \text{if } (b_m, b_n) \in E \\ min\{c_{p_1}, .., c_{p_k}\} & \text{if } \exists p_r = \{b_m, w_1, .., w_k, b_n\} \\ & \text{in G(V,E)} \mid \forall \text{ k } w_k \in \text{W} \\ \infty & \text{otherwise} \end{cases}$$

where c_{p_r} is the cost of the r-th path.

In order to find the route among all the semantic relevant rooms which minimizes the cost function, classical Nearest Neighbour and Greedy algorithms are applied to solve the Asymmetric Travel Salesman Problem on the reduced connectivity graph. The proposed tour is not always an optimal path in the sense of minimum distance, but it allows for a visit of the museum consistent with user preferences.



(a) (b)

Fig. 6. (a) Connectivity Graph of the whole environment; (b) Reduced Connectivity Graph of rooms containing relevant findings

V. CiceRobot at Work

The proposed architecture has been tested in the Archaeological Museum of Agrigento. As said, CiceRobot is able to guide visitors both in a prearranged tour and in an interactive tour, built in itinere depending on the user interaction.

When the visitor inserts a query a corresponding list of documents and findings is returned. This information is the start point to generate and simulate the plan of the mission task. The robot starts the visit giving some information about the museum. When the robot reaches one of the selected windows, it stops and gives the information previously retrieved. If CiceRobot is not able to reach one of the selected windows because of the presence of some visitors, it continues the visit and reschedules the skipped window. Let us consider a complete experiment. Supposing the visitor is in front of a sculpture of Kore and she asks to the robot: "Who is Kore?". The system suggests to expand the query adding the semantically nearest terms: "Persefone", "Demetra" and "Ade". If Demetra is selected, also findings not lexically related to the original query, e.g. "Demetra's Head", are proposed to the user (see right side of Fig. 7). In this case the semantic closeness values (cosine) of the retrieved findings respectively are: 0.92, 0.87, 0.82. Depending on the user selection, these values are updated. If the user selects the first (Kore's Head) and the second finding (Demetra's Head), semantic closeness values become: 0.92, 0.87, -1. The list of selected findings and the related semantic values are sent to the linguistic area. As a consequence, the default visit is updated: the selected findings are added, while the same number of default findings are removed according to their relevance in the museum (fixed in the default visit).

The Linguistic Area anchors the symbols of these findings to the corresponding knoxels in the CS allowing to update their semantic dimension. The Tour Building generates the initial plan that is progressively refined by simulation, before the real execution.

Figure 8 shows how the default visit is updated taking into account user preferences. In particular, the dashed line represents the default visit. According to user selection and to findings relevancy in the museum, Kore's Head and Demetra's Head (red bullet in figure) are added, while Dioniso and Kouros findings (crossed in figure) are removed. The continuous line represents the interactively built tour.

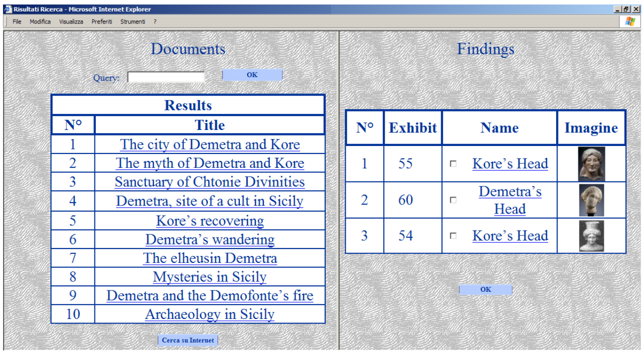


Fig. 7. Example of CiceRobot Interface

VI. CONCLUSIONS

We have presented a cognitive robot architecture based on the integration between subconceptual and linguistic computations through the introduction of the intermediate conceptual space. The architecture is organized in three computational areas. The subconceptual area is concerned with the processing of data coming from the robot sensors. In the linguistic area representation and processing are based on a semantic network formalism. This area is essentially the long-term memory of the robot. The conceptual area is intermediate between the subconceptual and the linguistic areas. Here, data is organized in geometric structures in terms of conceptual spaces. The paper also outlined the reciprocal roles of subconceptual computations, conceptual area representations and linguistic knowledge for behaviors planning based on a common semantic between human and robot. The architecture has been tested on an autonomous robot system on tasks related with guided tours in museum environment. We claim that the proposed architecture addresses the capacities which are generally addressed by an intelligent agent: the capability of representing itself and the external world, of imagining possible evolutions of itself and the world, of paying attentions to the relevant inner and outer events, of planning future actions and of evaluating situations and plans.

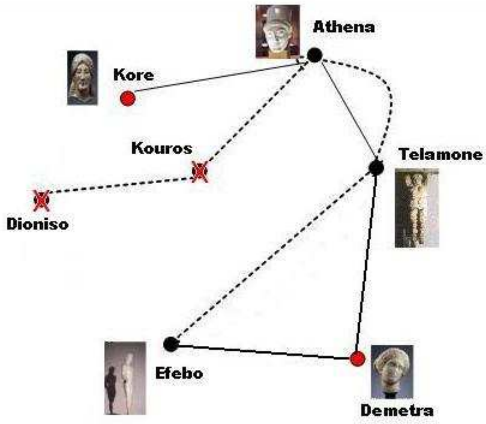


Fig. 8. Example of visit

REFERENCES

[1] A. Chella, M. Frixione and S. Gaglio, A cognitive architecture for artificial vision, *Artificial Intelligence*, vol. 89, 1997, pp. 73-111.
[2] A. Chella, M. Frixione and S. Gaglio, Understanding dynamic scenes, *Artificial Intelligence*, Vol. 123, 2000, pp. 89-132.
[3] P. Gärdenfors, *Conceptual spaces*, MIT Press, Cambridge, 2000.
[4] T.K. Landauer, P.W. Foltz and D. Laham, "An introduction to Latent Semantic Analysis", Discourse Processes,Vol. 25, No 2&3, 1998, pp 259- 284.
[5] B.V. Dasarathy, Nearest neighbor (NN) norms: NN pattern classification techniques, IEEE Computer Society Press, Los Alamitos, 1991.
[6] T. Mitchell, Machine Learning, McGraw Hill, New York, 1996.
[7] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin, 1995.
[8] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[9] W. Burgard, A. B. Cremers, D. Fox, D. Hhnel,G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, "Experiences with an interactive museum tour-guide robot", Artificial Intelligence, Vol. 114, No 1&2, 1999, pp. 3-55.
[10] A. Chella, S. Coradeschi, M. Frixione and A. Saffiotti, "Perceptual Anchoring via Conceptual Spaces", Proc. of the AAAI-04 Workshop on Anchoring Symbols to Sensor Data. AAAI Press (San Jose, CA), 2004.
[11] A. Chella, M. Liotta, I. Macaluso, "Cicerobot, A Cognitive Robot For Museum Tours", IASTED International Conference on ROBOTICS AND APPLICATIONS, October 31-November 2, 2005 Cambridge, USA.
[12] R.J. Brachman, J.C. Schmoltze, "An overview of the KL-ONE knowledge representation system", Cognitive Science, Vol. 9, No 2, 1985, pp.171-216.
[13] D.L. McGuinness, J.R. Wright, "An industrial strength description logic-based configurator platform", IEEE Intelligent Systems, Vol. 13, No 4, 1998, pp. 69-77.
[14] A. Maggio, A. Puglisi, S. Gaglio, G. Vassallo, G. Pilato, "Subsymbolic encoding of words", Lecture Notes in Artificial Intelligence Springer Verlag, No 2829, 2003, pp 449-461.
[15] S.T. Dumais, M.W. Berry, G.W. O'Brien, "Using linear algebra for intelligent information retrieval", SIAM Review, Vol. 37, No. 4, 1995, pp. 573-595.

Using Semantic Information for Improving Efficiency of Robot Task Planning

C. Galindo, J.-A. Fernández-Madrigal, and J. González
System Engineering and Automation Department
University of Málaga, (Spain)
{cipriano,jafma,gonzalez}@ctima.uma.es

A. Saffiotti
Center for Applied Autonomous Sensor Systems
Dept. of Technology, Örebro University
S-70182 Örebro, Sweden
{alessandro.saffiotti}@aass.oru.se

Abstract— **The use of semantic information in robotics is an emergent field of research. As a supplement to other types of information, like geometrical or topological, semantics can improve mobile robot reasoning or knowledge inference, and can also facilitate human-robot communication. These abilities are particularly relevant for robots intended to operate in everyday environments populated by people, which typically involve a great number of objects, places, and possible actions. In this paper, we explore a novel usage of semantic information: as an improvement for task planning in complex scenarios like the mentioned ones, where other planners easily find intractable situations. More specifically, we propose to first construct a "semantic" plan composed of categories of objects, places, etc. that solves a "generalized" version of the requested task, and then to use that plan for discarding irrelevant information in the definitive planning carried out on the symbolic instances of those elements (that correspond to physical elements of the world with which the robot can operate). Our results using this approach are promising, and have been compared to other existing approaches.**

Keywords- Semantic information, Abstraction, Generalization, Task Planning.

I. INTRODUCTION

The construction and study of representations of the real world is essential for the proper performance of mobile robots. The majority of approaches represent space and objects by only considering geometric information, for example building spatial maps (flat or hierarchical) with free and occupied areas [21],[17]. Although *geometrical* data is sufficient for solving a variety of robot tasks, other types of information are also useful: *topological* (to deal efficiently with large-scale maps), *hierarchical* (to deal efficiently with large amounts of information), and *semantic* (for performing more intelligently – from the point of view of a human being). In particular, semantic information could enable the robot to reason about the functionalities and characteristics of objects and environments [12], while topological symbols permit to communicate with humans using a proper set of terms and concepts [10].

The need to include semantic information in robot maps has been recognized for a long time [16],[5], but the integration of such information within spatial representations is still an emergent trend. Actually, most robots that incorporate provisions for task planning and/or for communicating with humans store implicitly some semantic information in their maps (e.g. [1],[20]), for example human-inspired classification of spaces (rooms, corridors, halls) or names of places and objects and the relations among them. However, these implicit approaches depend on the ability of the designers to capture the suitable set of semantic constraints and mechanisms.

In our previous work we have started to explore several ways in which the explicit use of semantic information may extend the robot capabilities:

- Semantic information enables a robot to infer new knowledge from its environment (e.g., to infer the type of a room according to the objects which are inside [12]).
- The use of both semantic and spatial information enhances human-robot communication by using concepts, terms, and reasoning understandable by people [10],.
- A robot can avoid misleading sensor readings by using semantic information. For instance, if the robot certainly knows that it is at a kitchen but its vision system detects a bathtub, it could discard this information since it knows that kitchens do not contain that kind of objects [12].

In this paper, we take one further step on using semantic information by exploring its benefits on improving symbolic task planning. We claim, as commented further on, that semantic information, when related appropriately to spatial information, can be exploited to help a robot to plan efficiently within large and/or complex worlds. In fact, some intractable problems under other planning approaches can become more tractable through the semantic support we proposed in this work.

In the robotics arena, though a great attention has been paid to path planning, task planning efficiency has been usually pushed to the background, most probably because of the simplicity of the scenarios considered so far in the field, or because of the limited forms of interaction between the robot and its environment.

This work has been supported by the Spanish Government under research contract DPI2005-01319, and by the Swedish KK Foundation.

However in the AI community, task planning efficiency has been largely studied [19],[22],[23],[24]. Some of their approaches have been borrowed in robotic applications, but up to our knowledge, only a few works have explicitly addressed the problem of planning efficiency for mobile robots [11],[9]. They usually rely on hierarchical structures upon topological representations to rid out irrelevant information to the task to be planned, achieving a significant speedup of the planning process.

Here we propose a novel hierarchical planning approach (called *SHPWA*) that exploits the semantic information managed by a mobile robot to improve its spatial-based task planning. Our semantic approach can be classified as a description logic system that organizes particular instances or world elements, say "my cup of coffee", into general categories, i.e. "Cup". This type of knowledge generalization produces a *taxonomic hierarchy* [14] (that we called here the *semantic hierarchy*).

The semantic hierarchy can provide the planning process (which is intended to construct a plan to solve the requested task using only spatial information) with information about the categories of world elements involved by the task at hand. Thus, we are able to discard irrelevant information for the spatial planning before it executes, reducing the computational effort. For example, if a servant robot is at a kitchen and it is commanded to "take my favorite fork" (let say *fork-1*), a possible solution could be "approach *shelf-1* and take *fork-1*". However, assuming a realistic environment, a kitchen may contain hundreds of objects and tens of distinctive places for navigation, which could prevent the planning process from achieving a successful result. In this paper, we claim that semantic information can help the planning process to reduce the search space: in our example, a fridge should not be considered for the task if the robot knows that forks are always in drawers; similarly, any spoon or knife could also be ignored although they can be found in a drawer. Our experiments show that using semantic information in this manner can help the robot to scale-up to environments containing thousands of objects.

In the rest of the paper, we describe our multi-hierarchical and semantic representation of the world (section II) and its use for improving the robot task planning process (section III). Section IV gives a comparison between our semantic-based hierarchical planner (SHPWA) and other non-semantic approaches (the flat planner Metric-FF [15], and the hierarchical planner HPWA [11]). Finally, some conclusions and future works are outlined.

II. SYMBOLIC REPRESENTATION OF THE WORLD

In our approach, we consider a mobile robot with a multi-hierarchical symbolic representation of its workspace [12]. This representation entails two hierarchies that represent the robot environment from two different perspectives: (i) a *spatial perspective*, that enables the robot to reliably plan and execute its tasks on existing objects, places, etc. (e.g., navigation, manipulation); and (ii) a *semantic perspective*, that provides it with inference capabilities (e.g., a *bedroom* is a *room* that contains a *bed*).

The spatial hierarchy contains symbols that represent particular elements of the environment, either perceived by the robot sensors or not. The ones that cannot be perceived are groupings of more detailed symbols and are useful for improving computational efficiency. The ones that can be sensed are created by means of the *anchoring technique* [11] that connects sensor data that refer to physical elements of the world, e.g. an image of my favorite fork, to particular symbols in the model, i.e. *fork-1*. This connection is represented by a data structure called *anchor* that includes a set of properties useful to re-identify the object, e.g., its color and position.

Figure 1 depicts our multi-hierarchical world representation, which includes the spatial and the semantic hierarchies. The *Spatial Hierarchy* arranges symbols in different hierarchical levels through abstraction of detail: (i) simple objects and distinctive places for navigation, (ii) the topology of the robot environment, and (iii) the whole environment represented by an abstract node. Additional intermediate levels could also be included.

The *Semantic Hierarchy* contains categories of the spatial symbols that represent particular elements of the world, i.e. *cup-1* in the spatial hierarchy is an instance of the category *Cup* in the semantic hierarchy. This categorization, represented in figure 1 as dotted lines, can be constructed by identifying particular properties of the correspondent anchors of the instances (i.e. a symbol is categorized as a cup if its anchor contains a perceptual image with a given size and shape), and/or through semantic inference, as presented in [12]. The semantic hierarchy may also model relations between categories, representing semantic knowledge like for instance that *Cups* "are usually on" *Tables*.

We manage this multi-hierarchical structure by using a mathematical model based on graphs called *MAH-graph* model [8], which has proved its suitability in reducing the computational effort of robot operations such as path-search [9] or symbolic task planning [11]. However, the semantic hierarchy can also be modeled by employing standard AI languages, like the *NeoClassic* language [18], in order to provide the robot with inference capabilities.

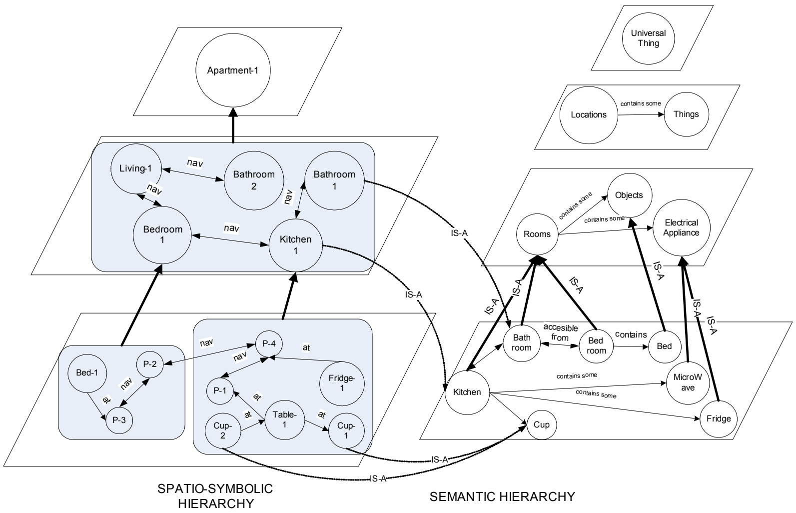


Figure 1. An example of the spatial and semantic hierarchies. On the left, spatial information represents the robot environment at different levels of detail. Shadowed regions indicate the set of symbols abstracted to the next upper hierarchical level. On the right, semantic information that models categories of spatial symbols and relations between them. This categorization starts in the links from symbols of the spatial hierarchy to categories in the semantic hierarchy (dotted lines). For clarity sake not all links and connections are shown in the figure, nor is anchoring.

A. The Spatial Hierarchy

The Spatial-Symbolic Hierarchy contains spatial and metric information from the robot environment. This model is based on *abstraction of detail*, that helps to minimize the information required to plan tasks by grouping/abstracting symbols in complex and high-detailed environments.

Symbols of this hierarchy are created by anchoring. On the one hand, laser-based gridmaps are anchored to symbols that represent distinctive places for navigation and open spaces [7], [3], on the other hand a visual pattern recognition system is employed to include information about the objects perceived by a camera mounted on the robot. When an object is detected that matches certain properties, i.e. a particular colour or shape, a new symbol is added to the model[1].

Symbols in both hierarchies are represented by vertexes which are interconnected through edges that represent relations between them. For instance, in the spatial hierarchy, two symbols that represent locations can be connected through a *navigability* edge, while a *located-at* edge models the relation between objects and a location (see figure 1).

Vertexes from the spatial hierarchy are abstracted into upper levels of the hierarchy to represent the topology of the environment, that is, a more general and less detailed representation of it. Different criteria can be adopted in order to construct these upper levels. We consider here grouping symbols according to their geometrical position and the normal distribution of human-like environments (places-rooms-apartments-buildings, etc.), though other techniques can be employed to construct hierarchies meeting other requirements, for instance for constructing hierarchies that improve the task planning process [9],[13].

B. The Semantic Hierarchy

The Semantic Hierarchy models semantic knowledge about the robot environment. All categories in this hierarchy are "refinements" of a common ancestor called *Universal Thing*, at the top level. Different sub-categorizations can be developed until reaching the lowest level that contains the most specific categories (kitchen, bedroom, cup, fridge, fork, etc.).

In our work, we incorporate semantic information within the multi-hierarchical model through semantic networks, as the one depicted in figure 2, although other mechanisms can be considered, like the *NeoClassic* system for knowledge representation and reasoning [18]. Regardless of the considered representation, the model should permit us to represent constraints or relations between categories, like for example the fact that a kitchen must have at least one fridge. The following is an example of how the category "kitchen" can be defined in the NeoClassic language. Intuitively, a kitchen is a room that has a stove, a fridge and a dish-washer, but does not have a bed, a bathtub, a sofa or TV set:

```
(createConcept Kitchen
(and Room
(atLeast 1 stove)(atLeast 1 fridge)
(atLeast 1 dish-washer)(atLeast 1 kitchen-furniture)
(and (atMost 0 bathtub) (atMost 0 sofa)
(atMost 0 bed) (atMost 0 tvset))))
```

The semantic network depicted in figure 2 shows the lowest level of the semantic hierarchy considered in our experiments.

[1] Since image recognition is out of the scope of our work we focus on recognizing simple objects, i.e., boxes, based on a particular shape and colour [12].

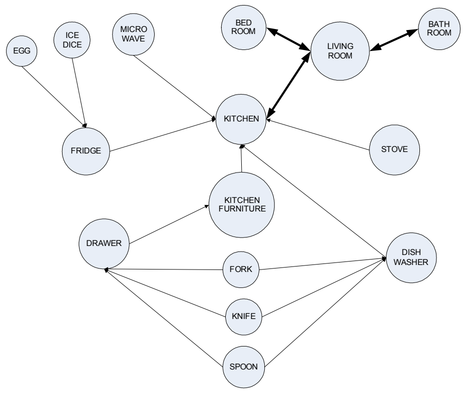


Figure 2. Example of the lowest level of a semantic hierarchy for an apartment scenario. Thick arcs indicate "can be accessed from", while thin ones indicate "can be found in".

As it will shown in the next section, a planning process able to cope with this semantic information could decide in advance the categories whose instances may be necessary for solving a given task, discarding the rest. Apart from the use of this information to guide task planning, it could also allow a mobile robot to perform some other types of inference, as described in [12]. For instance, if we know that "room-1" is a room that contains "obj-1", which is a fridge, then we can infer that "room-1" is a kitchen.

III. THE PLANNING PROCESS

Our semantic planning process is based on a previous work on hierarchical planning called HPWA (*Hierarchical Planning through World Abstraction* [11]). HPWA runs a given planner (the so-called *embedded planner*), like for instance Metric-FF [15], using the information stored at different levels of the spatial hierarchy to improve planning efficiency in large and complex domains – including those that are not large-scale but contain a high number of objects and/or relations).

Broadly speaking, HPWA receives as input a specification of the goal to achieve specified using symbols of the spatial hierarchy that belong to the same hierarchical level (typically the ground level). First, HPWA abstracts the goal to the upper levels, until the goal loses its meaning or becomes trivial (please consult the details in [11]). Then, it solves the requested task (the goal) specified at the highest level of abstraction of detail that has been reached. That abstract plan is used to discard irrelevant spatial information at the next lower level of the hierarchy (more detailed) by discarding the elements that are not involved in the abstract plan plus all the elements of lower hierarchical levels that abstract to them. This process is repeated until the level of the hierarchy where the requested task was originally specified is reached, providing a plan made of simple actions that the robot can carry out if all the elements of the task are appropriately anchored. In a typical real-world environment, which may contain hundreds of objects and distinctive places for navigation, the reduction in computational cost achieved by HPWA can be very important, as demonstrated in [11].

In this paper, we improve the computational efficiency of HPWA by using semantic information. The new approach is called SHPWA (for *Semantic HPWA*) and requires the use of a multi-hierarchical model like the one described in section II: the spatial hierarchy will serve for planning tasks as described before (tasks that contain real, spatial elements of the world on which the robot can operate), while the semantic hierarchy will provide support for planning categorical tasks (those that contain semantic symbols, that is, categories of objects, places, etc.). Broadly, the SHPWA process consists of two executions of the HPWA, as follows:

1) A task is requested for planning, for instance "take my favorite fork". This task is specified as a goal to achieve (a state of the world that have to be reached) using only symbols, i.e. *fork-1*, from some hierarchical level of the spatial hierarchy. Typically, that level is the ground level of the hierarchy and therefore all the symbols are anchored.

2) All the symbols in the goal are translated through the semantic links ("is-a") that connect both hierarchies to categories in the semantic hierarchy. Thus, the task "take my favorite fork" is translated to "take FORK". For the sake of simplicity we assume that all of the spatial symbols can be translated in this way; if this operation cannot be done, we pass directly to step 5 and thus semantics has not helped planning.

3) HPWA is used in the semantic hierarchy for constructing a categorical plan (one that only contains categories of objects, places, etc.) that solves the requested task. For the "take FORK" goal, considering the semantic network depicted in figure 2 and assuming that the robot is at the living-room, the categorical plan would be: "Go from LIVING-ROOM to KITCHEN", "Go to KITCHEN-FURNITURE", "Open DRAWER" and "Take FORK". We assume that if a plan exists that solves the task, there will exist a corresponding categorical plan in the semantic hierarchy. If that is not true, we go to step 5 becoming a non-semantic task planning approach.

4) The set of all the semantic categories involved in the categorical plan indicates the set of particular spatial symbols (places, objects, etc.) that will be involved in the final plan that is to be constructed on the spatial hierarchy. Therefore, all the symbols in that hierarchy that do not correspond to categories involved in the categorical plan are discarded for the subsequent planning process. In our example, only distinctive places of the kitchen, and the living-room, drawers of the kitchen's furniture, and forks are considered discarding the rest of elements of the domain.

5) HPWA is used in the spatial hierarchy for constructing the final plan considering only instances of the relevant categories provided by the semantic plan. In our example, the final plan would be: "go from living-1 to living-2", "go from living-2 to living-3", "go from "living-3 to kitchen-1", "go from kitchen-1 to furniture-1", "open drawer-3", "unstack fork-3", "unstack fork-2", "take fork-1", which can be executed by the robot.

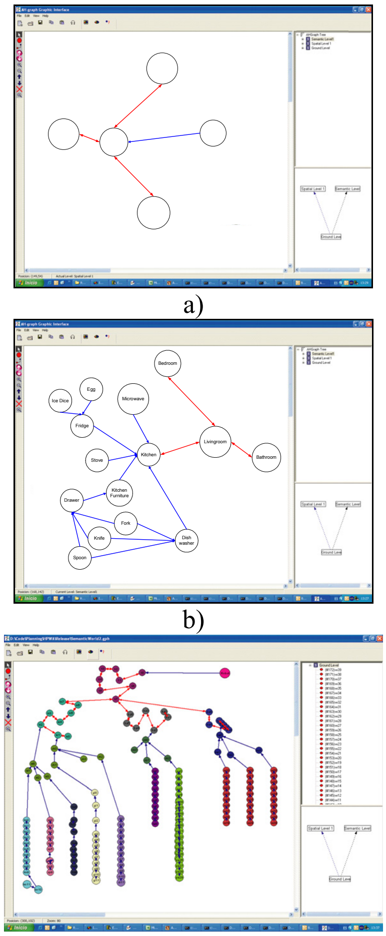
Step 1 assumes that all the symbols in a requested task have "is-a" links to the semantic hierarchy. This implies that the semantic hierarchy must be complete for all the elements of the world on which the robot will need to plan tasks, a reasonable assumption specially when the anchoring process, that is in charge of maintaining connections between real world elements and their spatial symbols usually provides information for the generation of their semantic categories [6], [12].

Note that step 3 does not involve a high computational effort, since although spatial information can grow rapidly through the exploration of the world, semantic one (the categories) remains generally bounded. Also notice that to carry out planning on the semantic hierarchy we need that relations between categories different from "is-a" (that is, those that do not serve to generalize) correspond to operational needs of the task. For example, if the task requested is "Put the *fork-1* in *drawer-3*", the corresponding goal will be translated to the semantic hierarchy into "A fork is in a drawer" and then planning will be performed on that goal. Thus, for carrying out that planning, the semantic hierarchy must contain relations between categories such as "forks are usually in drawers" (to find the fork), "drawers belongs to a piece of furniture" (to find the drawer), "kitchens and dining-rooms contains pieces of furniture", "kitchens are accessable from dining-rooms – and vice versa" (to take the fork to the kitchen if it is at the dining-room), etc. Further research must be conducted on the appropriate construction of these semantic hierarchies, which can be done manually (by a human programmer), automatically (by analyzing the spatial relations between objects and places to produce semantic information), or through a combination of both. In the experiments of this paper we have chosen the manual construction for simplicity.

IV. Experiments And Results

We have performed a number of experiments intended to verify the following hypothesis: using semantic information for task planning can reduce the complexity of the process, and therefore it allows us to scale up to larger domains than non-semantic planners. To test this hypothesis, we have used the SHPWA planner described in the previous section, and compared it to (non-semantic) HPWA. We also used a non-hierarchical planner (Metric-FF) for a baseline comparison. We have used a representation of a scenario with four rooms, containing many objects (grouped on piles) and distinctive places for robot navigation (see figure 3a). We have found that planning in such a scenario without semantic information may turn intractable even simple tasks.

We have run several experiments in which the number of objects and places have been gradually increased from 100 to 5000. For simplicity, in these experiments we have considered hierarchies with only two levels, though using additional hierarchical levels could improve even more planning efficiency [11]. It should be emphasized that the planned tasks were not actually executed on a robot: the goal of our experiments was to show the efficiency gained in the planning process by using a semantic-based planner, so executing the task was out of scope.



a)

b)

c)

Figure 3 Multi-hierarchical representation of a typical apartment scenario. a) First level of the spatial hierarchy that represents the different rooms and their connections. b) First level of the semantic hierarchy in which all the categories considered for this experiment and the relations captured by relations of particular instances are shown ("is accessable from" and "can be found at"). c) The spatial ground level for the simulated environment, with 100 world elements.

Figure 3 depicts the hierarchical model used in our experiments that represents an apartment. Figure 3a shows the first level of the *Spatial Hierarchy* in which topological information is grouped into rooms, and figure 3b shows the first level of the *Semantic Hierarchy* that categorizes the information of the ground level, which is shown in figure 3c.

In spite of the simplicity of the considered scenario, planning results using our semantic planning approach are promising. Several experiments have been conducted considering a random number of objects, places, and five different tasks (also chosen at random) consisting of taking an object (possibly manipulating other objects to take the desired one) and carrying it to a given location, i.e., "take *fork-1* to the *living- room-1*".

Figure 4 shows the average planning time for the set of random tasks varying the complexity of the environment (number of elements) with Metric-FF (planning only at the ground spatial level), HPWA (using all the levels of the spatial hierarchy, no semantics), and SHPWA. Although the behavior of each of the three planning approaches follows an exponential trend, the chart of figure 4 clearly demonstrates the benefits of using semantic information for planning. Also notice that in all cases the time of SHPWA is the shortest one, which proves that it actually alleviates the combinatorial explosion of the search involved in planning by discarding unnecessary objects for the task.

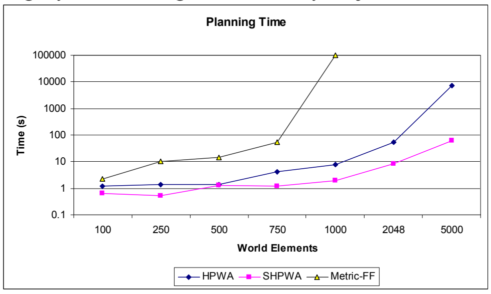


Figure 4. Task Planning comparison. Average time for planning five random tasks within random variations of the complexity of the considered simulated environment. Note the remarkable improvement achieved by SHPWA with respect to other non-semantic planners.

V. CONCLUSIONS AND FUTURE WORK

This paper has explored a novel way of using semantic information in mobile robot applications: improving robot task planning through semantics. The proposed approach envisages a multi-hierarchical model that coherently entails different sources of environmental information for robot operation. Within that model, one hierarchy (the spatial one) represents spatial symbols, some of them anchored to geometric information, while another hierarchy (the semantic one) represents semantic information arising from relations between particular instances of objects, places, etc. The information provided by semantic information is exploited for constructing a categorical plan to satisfy a given goal, which will serve to discard particular instances not involved in the solution, improving thus the overall task planning process. Planning experiences have demonstrated the benefits of using semantic information for planning tasks within complex scenarios in which a relative low number of objects makes other planners fail.

In the future, we plan to explore the human participation to help the robot to acquire more complex semantic information, the use of automatic procedures for that purpose, and to perform tests in which semantic-based planning is used in a physical robotic platform.

REFERENCES

[1] B.Bakker, Z.Zivkovic, and B.Krose. Hierarchical Dynamic Programming for Robot Path Planning. IROS 2005, pp.2756-2761.
[2] M. Beetz, T. Arbuckle, T. Belker, A.B. Cremers, D. Schulz, M. Bennewitz, W. Burgard, D. H¨ahnel, D. Fox, and H. Grosskreutz. Integrated Plan-Based Control of Autonomous Robots in Human Environments. *IEEE Intelligent Systems* 16(5):56–65, 2001.
[3] P. Buschka and A. Saffiotti. A Virtual Sensor for Room Detection. *IROS 2002*, Lausanne, CH, pp. 637-642.
[4] P. Buschka and A. Saffiotti. Some Notes on the Use of Hybrid Maps for Mobile Robots. *8th Int. Conf. on Intelligent Autonomous Systems* (IAS) 547–556.Amsterdam, 2004.
[5] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc of the IEEE Int Conf on Robotics and Automation*, pages 138–145, 1985.
[6] S. Coradeschi and A. Saffiotti. An Introduction to the Anchoring Problem. *Robotics and Autonomous Systems* 43(2):85-96, 2003.
[7] E. Fabrizi and A. Saffiotti. Extracting Topology-Based Maps from Gridmaps. In *Proc. of IEEE Int. Conf. on Robotics and Automation(ICRA)*, San Francisco, CA, pp. 2972–2978, 2000.
[8] J.A. Fernández-Madrigal and J. González, Multi-Hierarchical Representation of Large-Scale Space. Series on Micro-processor-based and Intell. Systems Eng. 24, Kluwer Academic Pub., 2001.
[9] J.A. Fernandez and J. Gonzalez. Multihierarchical Graph Search. *IEEE T. on PAMI* 24(1):103-113, 2002.
[10] C. Galindo and J. González and J.A. Fernández-Madrigal. A Control Architecture for Human-Robot Integration: Application to a Robotic Wheelchair. IEEE Transactions on Systems, Man, and Cybernetics Part B, vol 36, nº 5, pp.1053-1068, 2006.
[11] C. Galindo, J.A. Fernandez-Madrigal, and J. Gonzalez, Hierarchical task planning through world abstraction, IEEE Trans. on Robotics, 20 (2004), pp. 667–690.
[12] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernandez-Madrigal, and J. Gonzalez. Multi-Hierarchical Semantic Maps for Mobile Robotics. *IROS 2005,* pp. 3492-3497.
[13] C. Galindo. A Multi-Hierarchical Symbolic Model of the Environment for Improving Mobile Robot Operation. PhD Thesis, University of Malaga (Spain), 2006.
[14] Y. Gil. Description Logics and Planning. AI Magazine, 2005.
[15] Hoffmann J. and Nebel B., *The FF Planning System: Fast Plan Generation through Heuristic Search*. Journal of Artificial Intelligence Research, **14**, 2001 pp. 253-302.
[16] B. Kuipers.Modeling spatial knowledge.*Cognitive Science*,2, 1978.
[17] B. Lisiend, D. Morales, D. Silver, G. Cantor, I. Rekleitis and H. Choset. The Hierarchical Atlas. *IEEE Trans. On Robotics*, vol. 21, no.3, pp. 473-481, 2005.
[18] P.F. Patel-Schneider, M. Abrtahams, L. Alperin, D. McGuinness and A. Borgida. NeoClassic Reference Manual: Version 1.0. AT&T Labs Research, Art. Intell. Principles Res. Dept., 1996.
[19] E.D. Sacerdoti. Planning in a Hierarchy of Abstraction Spaces. Artificial Intelligence, **5**(2), 1974, pp. 115-135.
[20] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A 2nd generation mobile tour-guide robot. ICRA 1999, pp. 1999-2005.
[21] S. Thrun. Robotic Mapping: *A Survey. Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann*, 2002.
[22] M. Ghallab, D. Nau, and P. Traverso. Automated Planning. Morgan Kaufmann Publishers, May, 2004.
[23] J. Rintanen, J. Hoffmann. An Overview of Recent Algorithms for AI Planning. Künstliche Intelligenz, **2**(1), 2001, pp. 5-11.
[24] A. Botea, M. Enzenberger, M. Müller, and J. Schaeffer. Macro-FF: Improving AI Planning with Automatically Learned Macro-Operators. J. Artf. Intell. Research 24 (2005). pp. 581-621.

From Labels to Semantics: An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots

Óscar Martínez Mozos* Patric Jensfelt† Hendrik Zender‡ Geert-Jan M. Kruijff‡ Wolfram Burgard*
* University of Freiburg, Department of Computer Science, Freiburg, Germany
† Royal Institute of Technology, Center for Autonomous Systems, Stockholm, Sweden
‡ German Research Center for Artificial Intelligence (DFKI GmbH), Language Technology Lab, Saarbrücken, Germany
*{omartine, burgard}@informatik.uni-freiburg.de, †patric@nada.kth.se, ‡{zender, gj}@dfki.de

***Abstract*— We present an integrated approach for creating conceptual representations of human-made environments using mobile robots. The concepts represent spatial and functional properties of typical indoor environments. Our model is composed of layers which represent maps at different levels of abstraction. The complete system was integrated in a service robot which is endowed with laser and vision sensors for place and object recognition. It also incorporates a linguistic framework that actively supports the map acquisition process and is used for situated dialogue. In the experiments we show how the robot acquires the conceptual information and how it is used for situational and functional awareness.**

I. Introduction

Recently, there has been an increasing interest in robots whose aim is to assist people in human-like environments, such as domestic or elderly care robots. In such situations, the robots will no longer be operated by trained personnel but instead have to interact with people from the general public. Thus an important challenge lies in facilitating the communication between robots and humans.

One of the most intuitive and powerful ways for humans to communicate is spoken language. It is therefore interesting to design robots that are able to speak with people and understand their words and expressions. If a dialogue between robots and humans is to be successful, the robots must make use of the same concepts to refer to things and phenomena as a person would do. For this, the robot needs to perceive the world similar to a human.

An important aspect of human-like perception of the world is the robot's understanding of the spatial and functional properties of human-made environments, while still being able to safely act in it. For the robot, one of the first tasks will consist in learning the environment in the same way as a person does, sharing common concepts like, for instance, "corridor" or "living room". These terms can be used not only as labels but as semantic expressions that relate them to some complex object or objective situation. For example, the term "living room" usually implies a place with some particular structure, and includes objects like a couch or a television set. Moreover, a spatial knowledge representation for robotic assistants must address the issues involved with safe and reliable navigation control, with representing the space in a way similar to humans, and finally, with the way linguistic references to spatial entities are established in situated natural language dialogues.

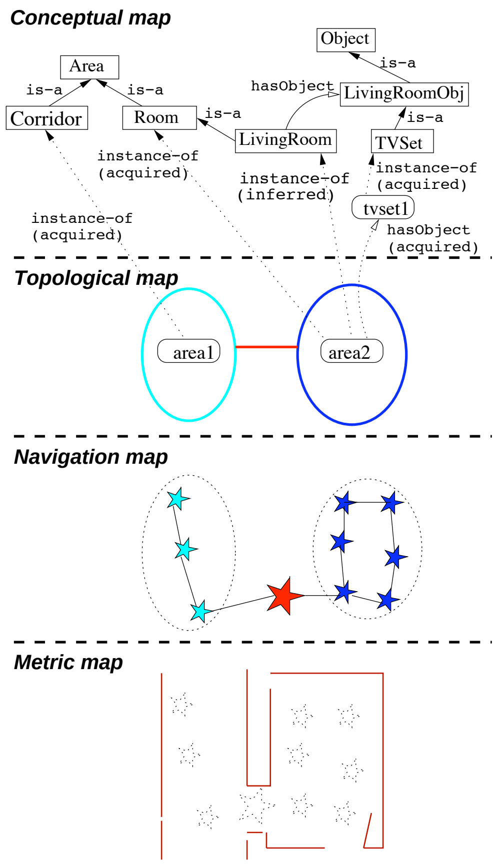


Fig. 1. An example of a layered spatial representation. Solid arrows indicate innate knowledge from the ontology. Dotted arrows refer to knowledge from the environment: asserted, acquired or inferred.

In this work we present an integrated approach for creating conceptual representations of human-made environments using mobile robots. The concepts represent spatial and functional properties of typical indoor environments. Our model is composed of layers containing maps at different levels of abstraction as shown in Fig. 1. The lower layers contain a metric map, a navigation map and a topological map, each of which plays a role in navigation and self-localization of the robot. On the topmost level of abstraction, the conceptual map provides a richer semantic view of the spatial organization, containing *acquired*, *asserted* and both *inferred* and *innate* conceptual-ontological knowledge about the environment. This model permits the robot to do spatial categorization rather than only instantiation.

The complete multi-layered representation is created in a semi-supervised map acquisition process, which is actively supported by a linguistic framework. This has been integrated into a cognitive system for mobile robots that is capable of conceptual spatial mapping in an indoor environment and that is endowed with the necessary abilities to conduct a reflected, situated dialogue about its environment.

The rest of the paper is organized as follows. In Section II, we present some related work. Section III describes our multi-layered conceptual spatial representation. The map acquisition process is outlined in Section IV. Situated dialogue is introduced in Section V. Section VI discusses how to achieve a notion of situational awareness using our conceptual representations. In Sections VII and VIII, we present implementation details and results respectively from an experimental evaluation of the integrated system. Finally, some concluding remarks are given in Section IX.

II. Related Work

Several approaches on mobile robotics extend metric maps of indoor environments with semantic information. The work by Diosi *et al.* [1] creates a metric map through a guided tour. The map is then segmented according to the labels given by the instructor. Martinez Mozos *et al.* [2] extract a topological semantic map from a metric one using supervised learning. Alternatively, Friedman *et al.* [3] use *Voronoi Random Fields* for extracting the topologies. In our system we use a similar approach to [2] for semantic classification.

Research in spatial representations has yielded different multi-layered environment models. Vasudevan *et al.* [4] suggest a hierarchical probabilistic representation of space based on objects. The work by Galindo *et al.* [5] presents an approach containing two parallel hierarchies, spatial and conceptual, connected through anchoring. Inference about places is based on objects found in them. Furthermore, the *Hybrid Spatial Semantic Hierarchy* (HSSH) is introduced by Beeson *et al.* [6]. This representation allows a mobile robot to describe the world using different representations each with its own ontology. Compared to these approaches our implementation uses human augmented mapping for collecting information. The communication with the robot is made entirely using natural language and dialogues. Moreover our conceptual representation comes from the fusion of acquired, asserted, and both inferred and innate knowledge.

There are more cognitively inspired approaches to robot navigation for conveying route descriptions from a technically naive user to a mobile robot. These approaches need not necessarily rely on an exact global self-localization, but rather require the execution of a sequence of strictly local, well-defined behaviors in order to iteratively reach a target position. Kuipers [7] presents the *Spatial Semantic Hierarchy* (SSH). Alternatively, the *Route Graph* model is introduced by Krieg-Brückner *et al.* [8]. Both theories propose a cognitively inspired multi-layered representation of the "map in the head", which is at the same time suitable for robot navigation. Their central layer of abstraction is the topological map. Our approach differs in that it provides an abstraction layer that can be used for reference resolution of topological entities.

A number of systems have been implemented that permit a robot to interact with humans in their environment. Rhino [9] and Robox [10] are robots that work as tour-guides in museums. Both robots rely on an accurate metric representation of the environment and use limited dialogue to communicate with people. The robot BIRON [11] is endowed with a system that integrates spoken dialogue and visual localization capabilities on a robotic platform similar to ours. This system differs from ours in the degree to which conceptual spatial knowledge and linguistic meaning are grounded in, and contribute to, situational awareness.

III. Multi-Layered Conceptual Mapping

The aim of this work is to generate spatial representations that enable a mobile robot to conceptualize human-made environments similar to the way humans do. These concepts correspond to spatial and functional properties of typical indoor environments. Following findings in cognitive psychology [12], we assume that topological areas are the basic spatial units suitable for situated interaction between humans and robots. We also proceed from the assumption that the way people refer to a place is determined by the functions people ascribe to that place and that the linguistic description of a place leads people to anticipate the functional properties or affordances of that place. At the same time, the constructed maps must allow for safe navigation and reliable self-localization of the robot. Considering these ideas, our final representation model is divided into layers, each representing a different level of abstraction. Each individual layer is important for the overall system because each layer serves a specific purpose. Starting from sensory input (laser scanner and odometry), a metric map and a navigation map representing traveled routes are constructed. On the basis of detected doorways, a topological partitioning of the navigation map is maintained. All these layers play a crucial role for the robot control systems. The conceptual map provides a conceptual abstraction layer of the lower layers. In this layer, spatial knowledge, innate conceptual knowledge and knowledge about entities in the world stemming from other modalities, such as vision and dialogue, are combined to allow for symbolic reasoning and situated dialogue. Fig. 1 depicts the four layers of the conceptual spatial representation.

A. Metric Map

The first layer of our model (Fig. 1, bottom) contains a metric representation of the environment in an absolute frame of reference. The geometric primitives consist of lines extracted from laser range scans. Such lines typically correspond to walls and other flat structures in the environment. The complete metric map is created by a mobile robot using *Simultaneous Localization and Mapping* (SLAM) techniques. The metric map is created online as the robot navigates around the environment based on the

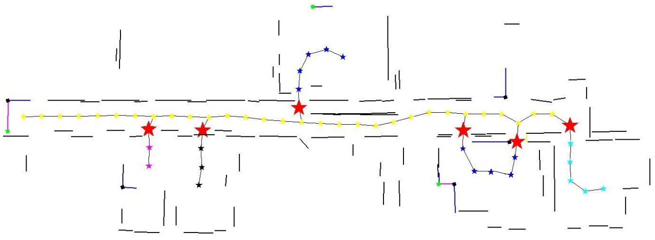


Fig. 2. The metric map is represented by lines. The navigation map is visually represented by the stars. Different colors represent different areas separated by doors, which are marked by bigger red stars.

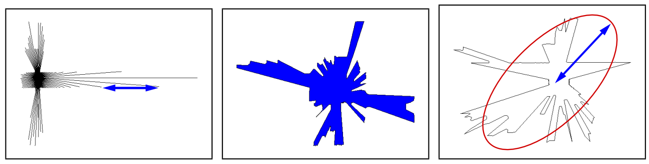


Fig. 3. Examples of features generated from laser data, namely the average distance between two consecutive beams, the perimeter of the area covered by a scan, and the mayor axis of the ellipse that approximates the polygon described by the scan. The laser beams cover a 360^o field of view.

same framework as in Folkesson *et al.* [13], which uses general representations for features that address symmetries and constraints in the feature coordinates to be added to the map with partial initialization. The number of dimensions for a feature can grow with time as more information is acquired. The basis for integrating the feature observations is the extended Kalman filter (EKF). An example metric map created using this method is shown in Fig. 2.

B. Navigation Map

The second layer contains the navigation map represented by a graph. This representation establishes a model of free space and its connectivity, i.e. reachability, and is based on the notion of a *roadmap* of *virtual free-space markers* [14], [15]. As the robot navigates through the environment, a marker (navigation node) is dropped whenever the robot has traveled a certain distance from the closest existing marker. The graph serves for planning and autonomous navigation in the known part of the environment.

We distinguish between two kinds of navigation nodes: place nodes and doorway nodes. Doorway nodes indicate the transition between different places and represent possible doors. They are detected and added whenever the robot passes through a narrow opening. Later, the status (open/closed) of a known door can be monitored using the laser scanner. Additionally, doorway nodes are assigned information about the door opening such as width and orientation.

Each place node is classified into one of two semantic labels, namely `Corridor` or `Room`, following the approach by Martinez Mozos *et al.* [2]. This method classifies the position of the robot based on the current scan obtained from the range sensor. The approach uses the AdaBoost algorithm to boost simple geometrical features into a strong classifier. Examples for typical features extracted from scans obtained in an office environment are shown in Fig. 3. The approach is supervised, which means that the robot must first be trained in an environment containing the semantic labels. As shown in [2] the training process does not have to be carried out in the same environment as the testing.

The approach for semantic classification assigns a label to each pose of the robot. To increase the robustness of the method, we classify each place node using the majority vote of the classification of the poses close to it. As explained before, a node is added when the distance to the previous node is greater than a threshold. We use this fact to store the classification of the last N poses of the robot in a buffer previous to adding the node. We then compute the majority vote of these last N poses and assign the final classification to the corresponding node.

C. Topological Map

The topological map divides the set of nodes in the navigation graph into areas. An area consists of a set of interconnected nodes (cf. Fig. 2). In this view, the exact shape and boundaries of an area are irrelevant. The set of nodes is partitioned on the basis of the door detection mechanism explained in the previous section. This approach complies with previous studies [12], [16], which state that humans segment space into regions that correspond to more or less clearly defined spatial areas. The borders of these regions may be defined physically, perceptually, or may be purely subjective to the human. Walls in the robots environment are the physical boundaries of areas. Doors are a special case of physical boundaries that permit access to other areas.

D. Conceptual Map

The conceptual map provides the link between the low-level maps and the communication system used for situated human-robot dialogue by grounding linguistic expressions in representations of spatial entities, such as instances of rooms or objects. It is also in this layer that knowledge about the environment stemming from other modalities, such as vision and dialogue, is anchored to the metric and topological maps.

Based on the work by Zender [17], our system is endowed with a commonsense OWL[1] ontology of an indoor environment (see Fig. 4) that describes taxonomies (*is-a* relations) of room types and typical objects found therein through *has-a* relations. These conceptual taxonomies have been handcrafted and cannot be changed online. However, instances of the concepts are added to the ontology during run-time. Through fusion of *acquired* and *asserted* knowledge – gathered in an interactive map acquisition process (cf. Section IV) – and through the use of the *innate conceptual* knowledge, a reasoner[2] can *infer* information about the world that is neither given verbally nor actively perceived. This way linguistic references to spatial areas can be generated.

[1] `http://www.w3.org/TR/owl-guide/`
[2] `http://www.racer-systems.org`

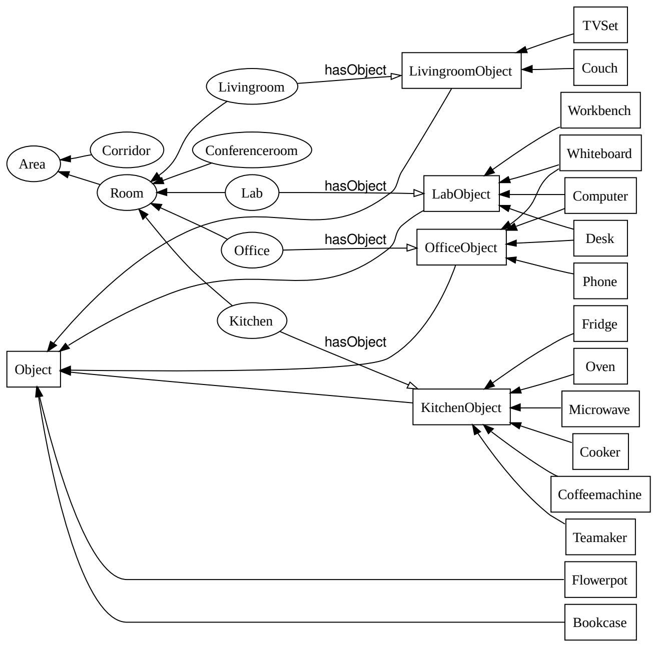


Fig. 4. Illustration of a part of the commonsense ontology of an indoor office environment. Solid arrows denote the taxonomical `is-a` relation.

1) Acquired Knowledge: While the robot moves around constructing the metric and topological maps, our system derives higher-level knowledge from the information in these layers. Each topological area, for instance, is represented in the conceptual map as an ontological instance of the type `Area`. Furthermore, as soon as reliable information about the semantic classification of an area is available, this is reflected in the conceptual map by assigning the area's instance a more specific type of either `Room` or `Corridor`. Information about recognized objects stemming from the vision subsystem is also represented in the conceptual map. Whenever a new object in the environment is recognized, a new instance of the object's type, e.g. `Couch`, is added to the ontology. Moreover, the object's instance and the instance of the area where the object is located are related via the `hasObject` relation. This process is shown in Fig. 1.

2) Asserted Knowledge: During a guided tour with the robot, the user typically names areas and certain objects that he or she believes to be relevant for the robot. Typical assertions in a guided tour include "You are in the corridor," or "This is the charging station." Any such assertion is stored in the conceptual map, either by specifying the type of the current area or by creating a new object instance of the asserted type and linking it to the area instance with the `hasObject` relation.

3) Innate Conceptual Knowledge: We have handcrafted an ontology (Fig. 4) that models conceptual commonsense knowledge about an indoor office environment. On the top level of the conceptual taxonomy, there are the two base concepts `Area` and `Object`. `Area` can be further partitioned into `Room` or `Corridor`. The basic-level subconcepts of `Room` are characterized by the instances of `Object` that are found there, as represented by the `hasObject` relation.

4) Inferred Knowledge: Based on the knowledge representation in the ontology, our system uses a description-logics based reasoning software that allows us to move beyond a pure labeling of areas. Combining and evaluating acquired and asserted knowledge within the context of the innate conceptual ontology, the reasoner can infer more specific categories for known areas. For example, combining the acquired information that a given topological area is classified as a room and contains a couch with the innate conceptual knowledge given in our commonsense ontology, it can be inferred that this area can be categorized as being an instance of `LivingRoom`. Conversely, if an area is classified as a corridor and the user shows the robot a charging station in that area, no further inference can be drawn. The most specific category the area instantiates will still be `Corridor`.

Our method allows for multiple possible classification of any area because the main purpose of the reasoning mechanisms in our system is to facilitate human-robot interaction. The way people refer to the same room can differ from situation to situation and from speaker to speaker, as reported by Topp *et al.* [18]. For example, what one speaker prefers to call the kitchen might be referred to as the recreation room by another person. Since our aim is to be able to resolve all such possible referring expressions, our method supports ambiguous classifications of areas.

IV. Interactive Map Acquisition

The multi-layered representation is created using an enhanced method for concurrent semi-supervised map acquisition, i.e. the combination of a user-driven supervised map acquisition process with autonomous exploration discovery by the robot. This process is based on the notion of *Human-Augmented Mapping*, as introduced by Topp and Christensen [19]. We additionally use a linguistic framework that actively supports the map acquisition process and is used for situated dialogue about the environment (see Section V).

The map can be acquired during a so-called guided tour scenario in which the user shows the robot around and continuously teaches the robot new places and objects. During such a guided tour, the user can command the robot to follow him or instruct the robot to perform navigation tasks. Our system does not require an initial complete guided tour. It is also possible to incrementally teach the robot new places and objects at any time the user wishes. With every new piece of information, the robot's internal representations become more complete. Still, the robot can always perform actions in, and conduct meaningful dialogue about, the aspects of its environment that are already known to it.

Whenever the user gives an assertion about areas in the environment or objects found therein, the robot updates the conceptual map with the asserted information. The concurrent constructions of the metrical map and the topological abstraction level propagate information in a bottom-up manner. Together with the laser-based area classification, these pieces of information lead to an update of the conceptual map with acquired knowledge.

Following the approach by Kruijff *et al.* [20], the robot can also initiate a clarification dialogue if it detects an inconsistency in its spatial representation, illustrating the mixed-initiative capabilities of the dialogue system.

V. Situated Dialogue

In this section, we will present the linguistic methods used for natural language dialogue with a robot. We will also address the role of dialogue for supervised map acquisition and task execution.

On the basis of a string-based representation that is generated from spoken input through a speech recognition software, the Combinatory Categorial Grammar (CCG) parser of OpenCCG[3] [21] analyzes the utterance syntactically and derives a semantic representation in the form of a Hybrid Logics Dependency Semantics (HLDS) logical form [22]. The dialogue system mediates the content from the speech input to the mapping or navigation subsystem in order to initiate the desired action of the robot or to collect pieces of information necessary to generate an answer. The generated answer string is then generated by the OpenCCG realizer and sent to a text-to-speech engine. The complete dialogue system is described in more detail in Kruijff *et al.* [23].

In the experiment of Section VIII, the user guides the robot around using a set of commands for initiating and stopping the interactive people following process and for instructing the robot with navigation commands to move near around. During this tour, the user augments the robot's internal map with assertions about the environment. In order to grasp the robot's understanding of its environment, the user has the possibility to ask the robot questions about the environment. The following examples contain HLDS representations of typical utterances in our scenario example:

(1) HLDS logical form of the utterance "This is the charging station."

$$
\begin{aligned}
&@_{\{B1:\mathsf{state}\}}(\mathbf{be}\\
&\quad \&\ \langle Mood\rangle\mathbf{indicative}\\
&\quad \&\ \langle Restr\rangle(T6 : thing\ \&\ \mathbf{this}\\
&\qquad \&\ \langle Proximity\rangle\mathbf{proximal})\\
&\quad \&\ \langle Scope\rangle(C3 : thing\ \&\ \mathbf{chargingstation}\\
&\qquad \&\ \langle Delimitation\rangle\mathbf{unique}\\
&\qquad \&\ \langle Number\rangle\mathbf{singular}))
\end{aligned}
$$

(2) HLDS logical form of the utterance "I am in a living room."

$$
\begin{aligned}
&@_{\{B9:\mathsf{state}\}}(\mathbf{be}\\
&\quad \&\ \langle Mood\rangle\mathbf{indicative}\\
&\quad \&\ \langle Restr\rangle(R2 : person\ \&\ \mathbf{I})\\
&\quad \&\ \langle Scope\rangle(I4 : region\ \&\ \mathbf{in}\\
&\qquad \&\ \langle Plane\rangle\mathbf{horizontal}\\
&\qquad \&\ \langle Positioning\rangle\mathbf{static}\\
&\qquad \&\ \langle Dir : Anchor\rangle(L1 : loc\ \&\ \mathbf{livingroom}\\
&\qquad\quad \&\ \langle Delimitation\rangle\mathbf{existential}\\
&\qquad\quad \&\ \langle Number\rangle\mathbf{singular})))
\end{aligned}
$$

[3] http://openccg.sourceforge.net

(3) HLDS logical form of the utterance "Follow me!"

$$
\begin{aligned}
&@_{\{F3:\mathsf{action}\}}(\mathbf{follow}\\
&\quad \&\ \langle Mood\rangle\mathbf{imperative}\\
&\quad \&\ \langle Actor\rangle(R7 : hearer\ \&\ \mathbf{robot})\\
&\quad \&\ \langle Patient\rangle(I2 : speaker\ \&\ \mathbf{I}))
\end{aligned}
$$

VI. Situational and Functional Awareness

We currently investigate how the information encoded in the multi-layered conceptual spatial representation can be used for a smarter, human- and situation-aware behavior. As one aspect of this, the robot should exploit its knowledge about objects in the environment to move in a way that allows for successful interaction with these objects. For instance, when following a person, the robot should make use of its knowledge about doors in the environment, such that it recognizes when the person wants to perform an action with the door. As actions that are performed in a doorway or with the door itself potentially require a wide space, e.g. for swinging or sliding open the door, for letting people pass, or for stepping past the door opening to grab the door handle, it is crucial that the robot adjusts its actions accordingly. A failure to understand such a situation could, for example, lead the robot to a position where it traps the user in the doorway that he or she was trying to close. In the experiment presented in this paper (see Section VIII), we opt for the robot to increase the distance it keeps to the user when it detects that the user approaches a door and to decrease it again when it detects that the user left the area. In this way, as the robot does not stop tracking and following the person, the people following behavior stays smooth and intuitive.

VII. System Integration Details

The complete system was implemented and integrated in an ActivMedia PeopleBot mobile platform (Fig. 5, left). The robot is equipped with a SICK laser range finder, which is used for the metric map creation, people following, and for the semantic classification of places. The place classification is based on a 360^o field of view (Section III-B). However our robot has only one laser at the front covering a restricted 180^o field of view. To solve this problem we follow the approach in [2] and maintain a local map around the robot, which permits us to simulate the rest of the beams covering the rear part of the robot. Additionally, a camera is used only for object detection. The detection systems uses SIFT features for finding typical objects like a television set, a couch or a bookcase. We recognize instances of objects and not categories [24]. The objects must be shown previously to the robot and learned by it (Fig. 5, right).

The communication with people was completely done using spoken language. The user can talk to the robot using a bluetooth headset and the robot replies using a set of speakers mounted on the mobile platform.

As an additional tool, we use an online viewer for the metric and navigation maps. The output of this program is composed of the lines extracted by our SLAM implementation extended to 3D planes to facilitate the visualization. The

Fig. 5. The left image shows the robot used during the experiment. The right images depict examples for for object detection: training couch image (top), detected couch image (bottom).

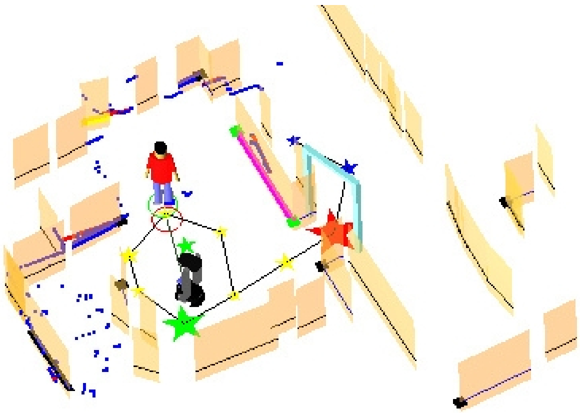


Fig. 6. Snapshot of the online viewer using during the experiment. The stars indicate the nodes in the navigation map. Blue for corridor, yellow for room, red for doorways and green for the actual position of the robot. Additionally, lines are extended to 3D planes and simulated doorways are drawn for facilitating the visualization. The person is drawn in the position detected by the people following software.

viewer shows the different nodes and edges used to construct the navigation map. Nodes corresponding to doorways are drawn bigger and with red color and with an associated doorframe (Fig. 6). Finally, the robot and the user are constantly shown in the positions where they are localized. The localization of the robot is calculated using SLAM [13], while the pose of the person is estimated using people tracking methods based only on laser readings [25].

The robot, being equipped with an onboard computer (850 MHz) connected to two built-in loudspeakers, runs the Player software[4] for control and access of the hardware, and the speech synthesis software[5]. The rest of the system runs on five laptops (1.8 GHz) interconnected using a wireless network. The first laptop is placed aboard the robot platform. It is connected to the onboard computer via an Ethernet crossover cable and to the rest of the system using its wireless adapter. This laptop runs the software for navigation, SLAM and people tracking. A second laptop runs the Windows operating system and is used for the real time speech recognition[6]. It is also placed on the robot platform in order to ensure a reliable bluetooth connection to the headset that recorded the user's voice commands. The recognized speech strings are sent to a third laptop, which runs the real-time dialogue processing and conceptual mapping subsystems. The fourth computer constantly classifies the current pose of the robot into a semantic class based on laser data. The last computer handles the viewer tool for debugging purposes. The communication between the different processes is established in a mixed environment using TCP/IP sockets and an OAA[7] framework. Fewer computers could have been used, but the setup was convenient as it allowed each subsystem developer to have his own computer.

VIII. EXPERIMENTS

In order to show all the functionalities explained in the previous sections, we carried out an experiment at the 7th floor of the CAS building at the Royal Institute of Technology in Stockholm. In this experiment the robot, together with a user, goes through different situations (or episodes). The complete experiment was carried out non-stop, i.e. we did not stop the robot or restart the system at any moment. The duration of the complete experiment was of approximately 6 minutes. Each of the episodes is explained in detail in the next sections and a video is available on the Internet[8]. The experiment was thought of as a test, and for this reason we "forced" some artificial situations to simulate possible real ones (e.g. the false doorway of Section VIII-B). A similar experiment was carried out in which the robot interacts constantly with the user and the environment for more than 30 minutes during a demo in the CoSy project[9]. In this case, the robot was presented to an audience while explaining its actions. Some of the episodes were repeated to clarify some questions. The robot again run with no interruptions or system problems. This led us to think that our implementation is quite robust and maybe can serve as basis for a long term service robot.

The idea of the experiments is to show how the robot learns its environment while interacting with a tutor. However, some previous knowledge is needed during this process. First, the robot needs an ontology representing the general knowledge about the environment. For this purpose, we use the ontology depicted in Fig. 4. Furthermore, the classification of places is based on previous general knowledge about the geometry of rooms and corridors, which is encoded in a

[4] http://playerstage.sourceforge.net/
[5] http://www.cstr.ed.ac.uk/projects/festival/
[6] http://www.nuance.com
[7] http://www.ai.sri.com/~oaa/
[8] http://www.dfki.de/cosy/www/media
[9] http://www.cognitivesystems.org

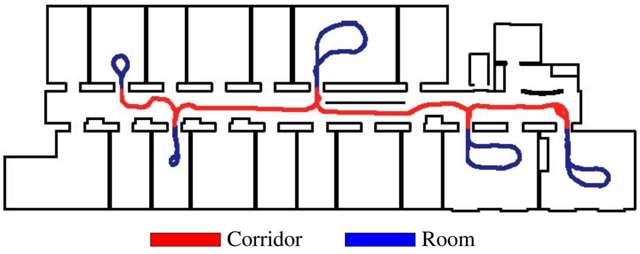


Fig. 7. Trajectory followed by the robot to train the classifier for distinguishing between corridor and room. The different places are depicted with distinct colors.

classifier based on laser readings as explained in Section III-B. The classifier is trained using examples of corridors and rooms from real environments as the one shown in Fig. 7. These two kinds of knowledge are independent of the environment used for testing, in the sense that the robot does not need to be physically present in the test environment to acquire the information. Finally, the robot has to recognize different objects, such as couches or TV sets, using vision. Because we do instance recognition rather than categorization, the objects we want to recognize must be presented to the robot before running the experiment. For this purpose, we position the robot in front of these objects, acquire a training image and label it with the corresponding term, which is added to a small database of objects and also included in the language systems for its posterior use.

A. *Episode 1: Waking Up*

The experiment starts in the corridor, where the robot is positioned close to the charging station. The user activates the robot and tells it that it is located at the charging station. The user then asks the robot to follow him. The robot drops markers (navigation nodes), which are classified as corridor. Then the person followed by the robot enters a room through a doorway. The door is recognized and the corresponding node is set. From this point the next nodes will be classified as a new area and correctly labeled as room.

B. *Episode 2: Clarification Dialogues*

In this episode we want to show the utility of the clarification dialogues. As explained in Section III-B, our door detection is simply based on detecting when the robot passes through a narrow opening. However, this alone will still lead to some false doors in cluttered rooms. Assuming that there are few false negatives in the detection of doors, we get great improvements by enforcing that it is not possible to change room without passing through a door. For example, while moving around in a room the robot may detect a narrow passage and falsely assume that a door was passed, putting a door label on that particular node. The robot continues to move around in the room and eventually reaches the nodes from before adding the false door. These nodes will then have different room labels, that is, the room has changed without passing a door. If this happens, an inconsistency is found and a clarification dialogue with the user is triggered.

Fig. 8. The user asks the robot: “Where is the charging station?”.

To test the former situation we put a bucket close to a table in the room creating an illusion of a doorway when using only the laser as sensor. The robot passes through this false doorway and comes back to a previously visited node. At this point the robot infers that there is an inconsistency in the map and initializes a clarification dialogue asking if there was a door previously. The user denies this fact and the map is updated accordingly. A more detailed explanation of the complete process of clarification dialogues for a similar situation is presented in Kruijff *et al.* [20].

C. *Episode 3: Inferring New Concepts*

In this episode we test how the robot infers new categorizations of places when discovering new objects. The goal is to use our SIFT-based object detector together with the laser-based place classification to detect simple objects and places. Then, using the inference on the office ontology as explained in Section III-D, the robot is able to come up with more specific concepts.

While staying in the room, the robot is asked for the current place and it answers with the indefinite description “a room”, which is inferred from the navigation nodes in the area. A majority vote among the nodes in the area is used in case the node classification is not unanimous. Then the robot is asked to look around. This command activates the vision-based object detection capabilities of the robot. The robot moves and detects a couch, and then a television set. After that, the user asks the robot for the name of the place. Because of the inference over the detected objects and places, the robot categorizes the place as a `Livingroom`. Note that previous to the detection of objects the same place was categorized as a `Room`. As a further test of the robot's classification it is asked where the charging station is located and correctly answers “it is in a corridor” (Fig. 8).

D. *Episode 4: Situational and Functional Awareness*

This episode shows the social capabilities of our robot. The robot must behave accordingly to the current situation, which in our case, is the opening of a door by the user (see Section VI).

Continuing with the experiment, the user asks the robot to follow him while he approaches a doorway. The robot knows from the navigation map where the doorway is and keeps a long distance to the user when he is near to the door. It then continues following the user by again decreasing its distance to him when he has passed the door. This action implies a certain degree of knowledge about social behavior, which is important if the goal is to create a robot that will live together with people.

E. Episode 5: Going to Objects

Finally, we show how the navigation map is used by the robot to come back to previously visited places.

After the door opening situation, the robot is asked to go to the television. The robot then navigates to the node where the television was observed. This functionality permits the user to command the robot to places without the need of giving concrete coordinates. It is also more powerful in the sense that the user may not know the concrete name of the place, but he can remember it as 'the room with a television". After that, the robot is commanded to go to the charging station. Again the robot follows the navigation map until it positions itself on the station, thus finishing the experiment.

IX. Conclusions

We presented an integrated approach for creating conceptual representations of human-made environments where the concepts represent spatial and functional properties of typical office indoor environments. Our representation is based on multiple maps at different levels of abstraction. The complete system was integrated and tested in a service robot which includes a linguistic framework with capabilites for situated dialogue and map acquisition. The experiments show that our system is able to provide a high level of human-robot communication and certain degree of social behavior.

Acknowledgment

This work was supported by the EU FP6 IST Cognitive Systems Integrated Project "CoSy" FP6-004250-IP.

References

[1] A. Diosi, G. Taylor, and L. Kleeman, "Interactive SLAM using laser and advanced sonar," in *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005.

[2] O. Martínez Mozos, A. Rottmann, R. Triebel, P. Jensfelt, and W. Burgard, "Semantic labeling of places using information extracted from laser and vision sensor data," in *IEEE/RSJ IROS Workshop: From Sensors to Human Spatial Concepts*, Beijing, China, 2006.

[3] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting the topological structure of indoor environments via place labeling," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

[4] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart, "Cognitive maps for mobile robots an object based approach," in *In Proc. of the IEEE/RSJ IROS 2006 Workshop: From Sensors to Human Spatial Concepts*, Beijing, China, 2006.

[5] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, 2005.

[6] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, and B. Stankiewicz, "Integrating multiple representations of spatial knowledge for mapping, navigation, and communication," in *Interaction Challenges for Intelligent Assistants*, ser. AAAI Spring Symposium, Stanford, CA, USA, 2007.

[7] B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, pp. 191–233, 2000.

[8] B. Krieg-Brückner, T. Röfer, H.-O. Carmesin, and R. Müller, "A taxonomy of spatial knowledge for navigation and its application to the Bremen autonomous wheelchair," in *Spatial Cognition*, ser. Lecture Notes in Artificial Intelligence, C. Freksa, C. Habel, and K. F. Wender, Eds. Springer Verlag, 1998, vol. 1404, pp. 373–397.

[9] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *ArtificialIntelligence*, vol. 114, no. 1–2, 2000.

[10] R. Siegwart and et al., "Robox at expo.02: A large scale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42, pp. 203–222, 2003.

[11] T. Spexard, S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. J. A. Kröse, "BIRON, where are you? - enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006, pp. 934–940.

[12] T. McNamara, "Mental representations of spatial relations," *Cognitive Psychology*, vol. 18, pp. 87–121, 1986.

[13] J. Folkesson, P. Jensfelt, and H. Christensen, "Vision SLAM in the measurement subspace," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005, pp. 30–35.

[14] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Academic Publishers, 1991.

[15] P. Newman, J. Leonard, J. Tardós, and J. Neira, "Explore and return: Experimental validation of real-time concurrent mapping and localization," in *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, Washington, D.C., USA, 2002, pp. 1802–1809.

[16] S. C. Hirtle and J. Jonides, "Evidence for hierarchies in cognitive maps," *Memory and Cognition*, vol. 13, pp. 208–217, 1985.

[17] H. Zender, "Learning spatial organization through situated dialogue," Master's thesis, Dept. of Computational Linguistics, Saarland University, Saarbruecken, Germany, 2006.

[18] E. A. Topp, H. Hüttenrauch, H. Christensen, and K. Severinson Eklundh, "Bringing together human and robotic environment representations – a pilot study," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[19] E. A. Topp and H. I. Christensen, "Tracking for following and passing persons," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, 2005.

[20] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Clarification dialogues in human-augmented mapping," in *Proceedings of the 1st ACM Conference on Human-Robot Interaction (HRI 2006)*, Salt Lake City, UT, USA, 2006.

[21] J. Baldridge and G.-J. M. Kruijff, "Multi-modal combinatory categorial grammmar," in *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, Budapest, Hungary, 2003.

[22] ——, "Coupling CCG and hybrid logic dependency semantics," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, PA, USA, 2002, pp. 319–326.

[23] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Situated dialogue and spatial organization: What, where. . . and why?" *International Journal of Advanced Robotic Systems, special section on Human and Robot Interactive Communication*, vol. 4, no. 2, March 2007.

[24] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *Int. Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91–110.

[25] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with a mobile robot using samplebased joint probabilistic data association filters," *International Journal of Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

ICRA-07 Workshop on
Semantic Information in Robotics
April 10, 2007, Italy

Acquisition of meaning through distributed robot control

Ricardo A. Téllez and Cecilio Angulo

Abstract— **We introduce a distributed neural network based architecture for the control of autonomous robots. This architecture is able to create a meaningful internal representation of the robot current situation directly grounded on its sensorimotor system. The representation is easily accessible from the outside and could be used for further deliberative purposes. An application example is provided for the garbage collector problem, where a robot must learn how to differentiate between garbage and walls, and attach those meanings to different sensor values.**

I. Introduction

In this paper, we address the problem of attaching meaning to a robot sensor state. The attachment of meaning to robot situations has been mainly done on a manual basis by the designers. In those cases, like for example in experts systems or in voice commands based robots like Aibo, the system operates over the syntax, and the semantic meaning is provided by a human who interprets the system answer, or includes it within the system itself [1]. This approach to semantic handling has been called *conventional functionalism*[2], and is characterized by a complete disentanglement between syntax and semantics. We are more interested, though, on the acquisition and maintenance of meanings by the artificial system itself. Systems equiped with this skill for automatic meaning acquisition are called *natural semantic* systems [3]. A natural semantic system creates and maintains its own meanings from its interactions with the environment.

Natural semantic systems are rare. However, there exists already some examples. For instance, Pierce and Kuipers [4] addressed the problem about a robot learning a model of itself and its environment without initial knowledge of the meanings of the sensors and actuators signals, and how all this knowledge could be used for prediction and navigation. A similar goal was achieved by Philipona et al. [5][6] where a robot was capable of inferring the external space to itself by studying the relations between motor commands and changes in the perception, otherwise called sensorimotor dependencies, by using a set of a priori unknown sensors and actuators. Another example of natural semantic system can be found in [7][8] where sensorimotor couplings were used to acquire the meanings of the robot sensors through sensory-invariance driven action. Finally, in [3], a robot learned to use its a priori unknown effector procedures to achieve its own internal goals.

All those works have in common that meanings are created through a sensorimotor coordination. The use of sensorimotor coordination for meaning acquisition is a real shift from the information processing approach used in most semantics free systems. This change from one approach to the other was proposed by Pfeifer and Scheier in [9][10] and by Nolfi in [11]. In their work, Pfeifer and Scheier view the problem of acquiring meaning as a problem of categorization. Categorization allows an agent immersed in the real world to make distinctions between different types of objects from the sensed values. When using an information processing approach, categorization is only seen as a mapping of sensory stimulation onto a library of stored internal representations. The sensorimotor approach instead, proposes the use of both sensor and effector in a coordinated way to perform the categorization. This approach states that both sensor and motor play an important part in the act of categorizing and by hence, in the acquisition of meaning.

In this paper we present a distributed architecture which allows a robot to automatically acquire the meaning of its sensory inputs, creating an internal representation of it. This representation is like an internal meaningful categorization of the robot situation, created through sensorimotor coordination. Furthermore, this categorization is directly accessible as the output of some modules, hence, it is suitable for its use by other modules. The rest of the paper continues with a description of the architecture employed (in section 2), and follows with an application of the architecture to the garbage collector problem (section 3). The results obtained from the resolution of the garbage problem are used to analyze the inner workings of the architecture and see how meanings are created (section 4). Section 5 discusses the results obtained, and section 6 concludes and points to future work.

II. Architecture description

We have created a distributed architecture for the control of autonomous robots, based on neural networks. It is called Distributed Architecture with Internal Representation (DAIR), and a description of its more relevant issues for this paper are included below. The main goal of this architecture is to allow the generation of complex behaviors in complex robots within the evolutionary robotics framework. Because of that, a complete modular distributed architecture was developed. The use of such degree of modularity allows the staged evolution of controllers for robots with several sensors and actuator in a process that we call progressive design. A complete description and comparison of the architecture against other evolutionary robotics architectures can be found in [12]. The description of the staged evolution process for progressive design is described in [13]. The application of the

Technical University of Catalonia, Avinguda Víctor Balaguer s/n, 08800 Vilanova i la Geltrú - Barcelona, Spain rtellez@lsi.upc.edu , cecilio.angulo@upc.edu

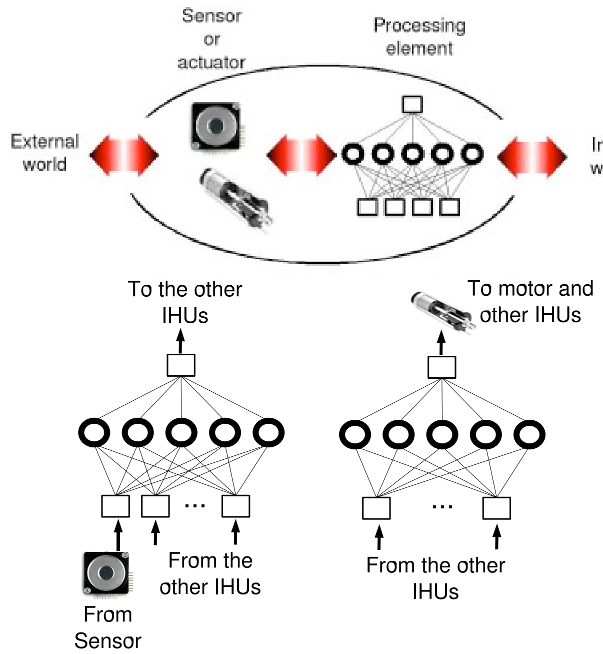


Fig. 1. IHU schematics (above), and connection schematics of the processing element to the associated device (sensor or actuator) (below).

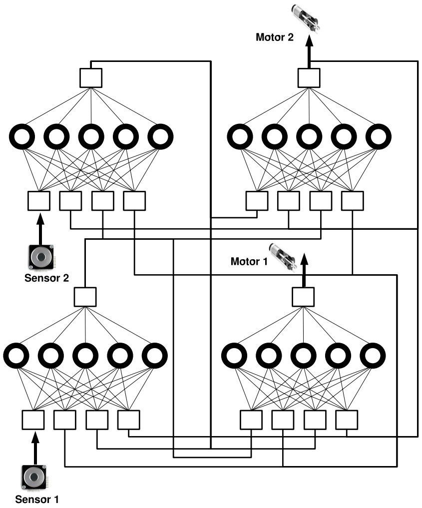


Fig. 2. Application of the DAIR architecture for the control of a simple robot composed of two sensors and two motors. Four IHUs are required.

architecture to a complex Aibo robot using staged evolution can be found in [14], [12].

The DAIR architecture is a distributed modular approach to autonomous robot control. Modularity is implemented by creating a small processing module around each of the robot sensors and actuators. Each module is created by what is called an Intelligent Hardware Unit (IHU) whose schematics is shown in figure 1.

Every IHU is composed of a sensor or an actuator and a processing element which processes the information of its associated device, that is, received sensor information for sensors, and commands sent to the actuator for actuators. It is said that the processing element is in charge of its sensor/actuator. This type of connectivity means that the processing element is the one that decides which commands must be sent to the actuator, or how a value received from a sensor must be interpreted. All IHUs are connected to each other, allowing to each IHU know what the other IHUs are doing. This implies that the processing element is also in charge of deciding what to communicate to the other elements as well as to interpret what the others are communicating.

Hence, the architecture allocates one module for each device. Eventhough each module is independent and perform its own program associated to its device, modules will still have strong couplings between each other. This type of modularity implies that the optimal solution for the control of one device by its IHU will in fact depend on the optimal solutions found by the other IHUs. This type of modularity where great couplings between modules exist has been called decomposable modular system [15], [16].

As a processing element, a neural network was selected. Neural networks are easily evolvable using evolutionary robotics procedures, and present several advantages like inmunity to noise, allow the progresive evolution of its weights, and present a graceful degradation. The type of neural network used will depend of the task to be solved. For instance, in [14] a simple FeedForward neural network with hidden units was used on a standing up behavior. In [12] a Continual Time Recurrent Neural Network was used for the generation of a walking behavior. In this paper a simple FeedForward net with no hidden units was used (see figure 4-bottom). The structure of a IHU can be seen in figure 1, and figure 2 shows how a complete neural controller would be constructed for a simple robotic system composed of two sensors and two actuators. It should be stated that when put several IHU together on a control task, each element has its own particular vision of the situation because each one is in charge of its own sensor or actuator. This means that there is no central coordinator. Each unit knows what the others are doing but needs to select an action for its actuator or sensor output, based on its knowledge of the global situation and the current state of its particular device.

Hence, a distributed coordination between all the elements is required which allows the whole robot perform the behavior required without the use of a central coordinator. In our case, this is accomplished through an evolutionary process using a neuro-evolutionary algorithm. Due to the fact that the evolutionary process has to evolve different ANNs for different roles on a common task, a co-evolutionary algorithm is required, that is, the simultaneous evolution of several nets with a common fitness. By using such kind of algorithm it is possible to teach to the networks how they must cooperate to achieve a common goal (i.e. the global robot behavior to implement), when every network has its own an different vision of the whole system.

The algorithm selected to evolve the nets is the ESP

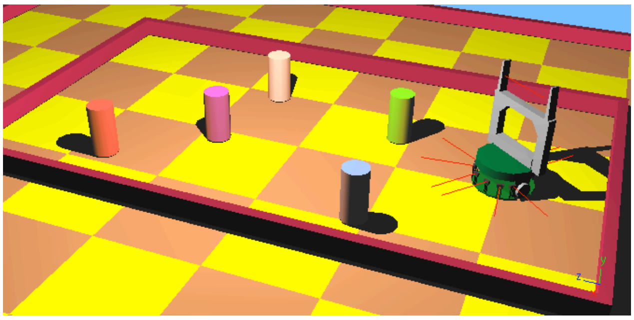


Fig. 3. Simulation of the garbage collector problem on Webots simulator

(Enforced Sub-Populations) [17][18], which has been proved to produce good results on distributed controllers [19]. A chromosome is generated for each IHU network, coding in a direct way the weights of the network connections, and the whole group of neural nets is evolved at the same time with direct interaction with the environment. The fitness function which guides the evolutionary process is created by the designer, depending on the problem that the robot has to solve.

III. Application to the Garbage Collector Problem

In order to test the theoretical approach presented in the previous section and see how meanings are created, a Khepera robot simulation was used as test bed. Experiments consisted of the implementation of the DAIR architecture for the control of a Khepera robot while performing a cleaning task. The selected test bed task is called the garbage collector, and follows the description given in [11]. In this task, a khepera robot is placed inside an arena surrounded by walls where it should look for any of the sticks randomly distributed on the space, grasp it, and take it out of the arena (figure 3). The garbage collector behavior requires that the robot completely changes its behavior based on a single sensor value change. When the robot does not carries a stick on the gripper, then its behavior has to avoid walls, look for sticks, approach them, and pick them up. When the robot carries a stick, its behavior has to change to the opposite, avoiding other sticks and approaching walls in order to release the stick out of the arena. This kind of test will allow us to see if the robot creates different classifications for the same object depending on the status of the gripper, or otherwise, the robot has only a single representation for the same object independently of its gripper state, since the object perceived would be the same in both cases.

A. Experiment setup

All the experiments reported for the Khepera robot were done on a simulator. As simulator, we selected the commercially available Webots simulator by Cyberbotics [20]. This simulator includes, among other things, the simulation of the Khepera gripper, which is the turret capable of grasping objects (see figure 3). The Khepera gripper is composed of an arm that can be moved through any angle from vertical to horizontal, and two gripper fingers that can assume an open or closed position. The gripper is also composed of a sensor that indicates the presence of an object between the fingers.

The robot has eight infrared sensors, six on the front and two on the back. For the resolution of the garbage collector problem only the six front sensors were used, as well as the gripper sensor. As actuators, the robot has two motors (left and right), but it is also possible to control the position of the gripper arm and the status of the gripper fingers (open or close). The control of the gripper is done by means of two procedures: the first procedure, when activated, moves the arm down, closes the gripper fingers and moves the arm up again, picking a stick up; the second procedure moves the arm down, opens the gripper fingers, and moves the arm up again, releasing the stick.

The same setup as in Nolfi's work was implemented for the garbage collector task. It is composed of a rectangular arena of 60x35 cm, surrounded by walls, and containing five garbage cylindric sticks. Each stick has a diameter of 2.3 cm and was positioned randomly inside the arena at every new epoch. In the same way, the robot was also randomly positioned on the arena at the beginning of each epoch.

Experiments consisted of 15 epochs of 200 time steps each, where an evolved controller was tested over the task. The duration of each time step was of 100 ms. Each epoch ended after the 200 steps or after a stick had been correctly released out of the arena.

The DAIR architecture implementation creates one IHU element for each device involved. There were eleven devices involved, thus eleven IHUs were created: an IHU for each of the infra-red sensors and the gripper sensor was created (seven in total), two IHUs for the left and right motors, and other two for the two gripper procedures. Each IHU was implemented by a feedforward neural net with eleven inputs, no hidden units, and one output.

The architecture was evolved using the evolutionary setup described above. A fitness function was created for the evolutionary process which rewarded controllers capable of releasing one stick out of the arena. Controllers that were able to only pick up one stick were also rewarded with a lower fitness.

$$fitness = \begin{array}{cc} 0.1 & if\ pick\ up\ stick \\ 1 & if\ stick\ released\ outside\ arena \\ 0 & if\ stick\ released\ inside\ arena \end{array}$$

Like in the original experiments made by Nolfi, a special mechanism was implemented which artificially added a stick in front of the robot each time it picked one stick up. The reason was to increase the situations where the robot encountered an obstacle in front of it while carrying a stick. One epoch lasted either 200 steps or until a stick was released outside the arena. Each controller was tested for 15 epoch per generation, obtaining the final fitness of the controller as the average fitness of all the 15 epochs. Each evolutionary process lasted for 1000 generations. Due to the stochacity of the method employed, the whole evolutionary process was

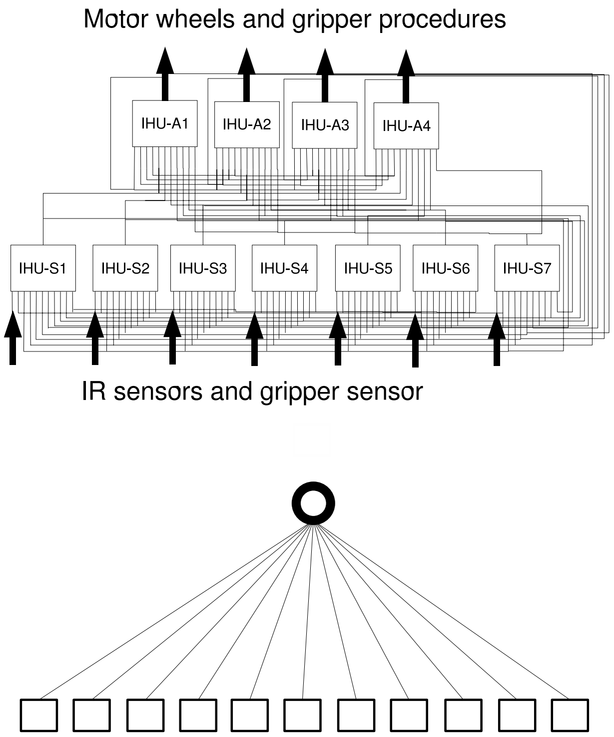


Fig. 4. Modular representation of the architecture implemented for the Khepera robot (top), and the neural network used for each IHU module (bottom)

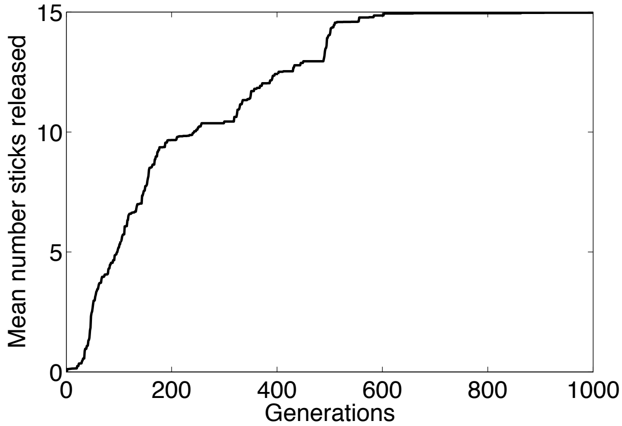


Fig. 5. Mean number of sticks (out of 15) correctly released outside the arena through generations. The curve represents the average in 10 different evolutionary processes.

performed ten times.

B. Results

After 1000 generations, 9 out of the 10 evolutionary runs evolved a maximal fitness behavior (15 sticks released out of 15 epochs), generating a distributed controller able to perform the garbage collector behavior[1]. Results presented in figure 5 show the evolution of the averaged fitness for those ten runs.

When the controllers obtained are observed on the simulator, we cannot appreciate significant differences in their behavior. All of them perform correctly the behavior of looking for sticks while avoiding walls, pick a stick up, and then release it outside the arena while avoiding other sticks. However, it happens in some special cases that the robot categorizes a stick as a wall, while not carrying a stick. This has not been considered as an error, since this type of wrong classification does not lead to any error in the global behavior. This situation was also observed in Nolfi's original experiments, and could have been avoided in both cases by complexifying the fitness function or by providing to the robot with such strange situations during the evolutionary process as was done with the stick which was put in front of the robot once it picked a stick up.

[1]video of the behavior obtained available at www.ouroboros.org/garbage_collector.html

IV. ACQUISITION OF MEANINGS

An analysis of the inner workings of architecture shows that the architecture makes use of the sensorimotor metaphor for the apprehension of meaning and its assignment to sensory states. This section will show the meanings generated by the architecture and how can they be accessed from the outside.

When analyzing the outputs of the evolved sensor IHU modules, we observe that they produce similar output patterns to similar situations. The sensor IHUs provided the same output values to different sensor values which corresponded to the same *conceptual* situation. This implies that the sensor IHUs were classifying all a bunch of different sensor states into the same conceptual category or meaning. The different categories or meanings can be accessed by using what we call the *state vector* of the robot at a given time step. The state vector is formed by the concatenation of the output values of the sensors IHUs at each time step, that is:

$$state\ vector = (IHU_{S1}, IHU_{S2}, IHU_{S3}, IHU_{S4}, IHU_{S5}, IHU_{S6}, IHU_{S7})$$

This state vector identifies the situation of the robot at that time step. Basically, it can be seen as a categorization of its current situation, or as an internal modeling of the outside world that the robot is experiencing at this particular moment. This internal representation at the IHU level contains the meaning of the situation, and that meaning is attached to the present sensor activity pattern. Changes in the values of the sensors did not change the state vector, unless a change in the situation of the robot, relevant for the task to solve, was produced. Changes from one state to another one are not instantaneous and involve a transient time where the IHUs exchange information and finally adopt the new state.

The internal representations that map the sensory stimulation to the category actually been experienced are automatically created by the evolutionary process while interacting with the environment. Therefore, the meanings are grounded into the robot experiences. This means that the actual states identified by the robot have a meaning for the robot. However, this meaning does not have to correspond to a human meaning, but a meaningful state for the robot for the task to be solved.

As will be seen below, for the garbage collector problem, the robot identifies only a few possible states as required for the solution of the task at hands, allowing it to reduce the huge number of possible sensor inputs and robot states to that few number of relevant ones. This means that a group of sensors values will always correspond to a unique single meaning or category. This represents a huge reduction from the high number of possible situations that raw sensed data provide. The internal states created by the system identify those states that have a real semantic value, and that value is grounded to the experiences of the robot.

For the garbage collector problem, there have been identified eight different internal states, each one corresponding to a meaningful situation for the robot. In order to identify the states that the robot evolved, some experiments where performed. Those experiments consisted of allocating the robot on a special situation, and then measure the values given by the sensor IHU modules until the situation changed (by means of the robot action). Special situations included putting the robot on free space, and putting the robot in front of a stick or a wall with different collision angles and distances. All situations were tested with and without carrying a stick.

By observing the graphics produced by the IHUs on each of the special situation experiments, we obtained the IHU output values presented in table 1. This table represents the state vectors obtained with one of the 9 controllers evolved. Values presented here were not clear and neat values, but small variations of the order of 0.05 were observed in the same state in different situations. Furthermore, the table represents the values obtained for only one of the 9 distributed controllers obtained. The same concept states were obtained for the other controllers, but their vectorial values were not the same, since the evolutionary process is of stochastic nature, which leads to the evolution of different vector values for the same conceptual states.

It follows a description of the identified states:

State a: This state is obtained when the robot does not carry a stick and does not detect anything. The robot is put in the middle of the arena and no obstacles are put besides it. After an initial transient time, the robot starts moving forward, assuming a stable state where the values of the IHUs outputs do not change at all, making the robot advance forward. This behavior ensures that the robot will eventually detect something, either the wall or a stick.

State b: This state is obtained when the robot carries a stick and does not detect anything. This situation is the same as in the previous state, but now the robot has a stick on its gripper. Basically, the state of the robot is the same as in the previous one, except that the IHU of the gripper sensor indicates that there is a stick on it.

State c: State obtained when the robot detects something but it does not know what it is (a wall or a stick). This state happens when the robot detects something with sensor E but it is not capable of classify what it is. This state, motivates a special response pattern in the motor IHUs that makes the robot turn over itself in order to allow sensors C and D detect the object, and help it to disambiguate the sensing information. Value v1 changes depending on the distance to the object.

State d: State observed when the robot does not carry a stick and it is in front of a wall. In this case, the robot realizes that there is a wall in front of it, so it starts a movement in order to avoid it.

State e: This state occurs when the robot does not carry a stick and it is in front of a stick. Now, the robot detects the stick and recognizes it as that. Therefore, it activates the pick-up procedure in order to pick the stick up. Value v2 changes depending on the distance to the object.

State f: State observed when the robot carries a stick and detects another stick. In this situation, the robot changes its behavior to avoid the detected stick. Strangely, this state is different from the state where the robot did not carry a stick and detected a wall. Value v3 changes depending on the distance to the object.

State g: This state is observed when the robot carries a stick and detects a wall. In this case, the robot categorizes the obstacle as a wall and then activates the releasing stick procedure.

Those observed states indicate that the DAIR architecture uses indeed the sensorimotor coordination metaphor in order to produce its categorization. The most clear example is the result obtained in *state c*, where the robot detects something but it can not identify what it is. This situation indicates that the robot is having perceptual aliasing. Its strategy is to move itself into a more convenient position which provides it with a more convenient sensor input that allows it to determine what it is in front of. This type of behavior is just what has been called as *active perception* [21] or as we have being calling it during this paper, sensorimotor coordination.

V. Discussion

We have shown how a distributed architecture can create and use meaningful representations for the resolution of the garbage collector task. However, it can be argumented that this representation was also generated on the original experiments by Nolfi, because he was able too to solve the garbage problem. The advantage of the DAIR architecture is that the categorization created is directly accessible to an observer external to the networks, that is, the meanings not internally coded in the network weights. This means that it is possible to direct access the present situation of the robot from a conceptual point of view by just looking the IHU sensor outputs. This type of direct access to the generated meanings may not be necessary in biological intelligent systems, but scientists feel more confortable when such differentiation is possible because allows an easier understanding of the whole process. Furthermore, it may help in the maintenance

Sensors	IHU A	IHU B	IHU C	IHU D	IHU E	IHU F	IHU Gripper
State a	0.06	0	1	1	0.97	0	0.1
State b	0	0	1	1	0.99	0	1
State c	0.06	0	1	1	v1	0	0.1
State d	0.06	0	0	0	0	0	0.01
State e	0.6	0	v2	0	0.19	0	0.01
State f	0.6	0.96	v3	0	0	0	1
State g	0.01	1	0.05	0.17	0	0	0.97

TABLE I

TABLE CONTAINING THE OUTPUT VALUES OF EACH IHU SENSOR FOR THE INTERNAL STATES CREATED. LETTERS A TO F INDICATE EACH OF THE IR SENSORS FROM LEFT TO RIGHT.

of a correspondence between syntax and semantics. This could be achieved, by accessing to the meanings created by a more deliberative superior layer, which would use them to (syntactically) reason about its situation, propagating in this way the robot acquired meanings to more syntactic processes.

From another point of view, we can see the actuation of the architecture as an extractor of meaningful events which are relevant for the resolution of the task. We have seen that the architecture is capable of converting a continuous flow of sensor data into a discrete number of meaningful situations. We will call this situations events. A new event is generated each time that the situation for the robot changes. And the situation changes when the robot itself *thinks* that the new sensory flow corresponds to something really different from previous situation. In fact it creates a categorization of experiences useful for the task at hands. This behavior is similar to the ARAVQ event extractor algorithm [22], with the difference that the ARAVQ extracts the events from the information gathered by a robot that already knows how to solve the task. Instead, the introduced architecture learns to extract the events while learning the resolution of the task.

VI. CONCLUSION

We have presented a distributed architecture able to control a robot through sensorimotor coordination. The architecture creates its meanings from interaction with the environment, and uses those meanings to classify and solve a garbage collector task. Future work will continue with this bottom-up approach, exploring how to use the state vector generated by the sensor modules to integrate deliberative processes which decide depending of the current situation of the robot, as it is perceived by the robot itself.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Olivier Michel for his support in the development of the garbage collector simulation, and Diego Pardo for insightful suggestions. This work has been partially founded by the Spanish Ministry of Education and Science ADA project number DPI2006-15630-C02-01.

REFERENCES

[1] D. Dennett, "Do-it-yourself understanding," in *Brainchildren*. MIT Press, 1998. [Online]. Available: http://pp.kpnet.fi/seirioa/cdenn/doityour.htm

[2] N. Block, "What is functionalism?" Published on web from a revised version of the entry on functionalism in The Encyclopedia of Philosophy Supplement, Macmillan, 1996.

[3] P. Cohen and C. Beal, "Natural semantics for a mobile robot," University of Massachusets, Department of Computer Science, Tech. Rep. 00-59, 2000.

[4] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence Journal*, vol. 92, pp. 169–229, 1997.

[5] D. Philipona, J. K. O'Regan, and J.-P. Nadal, "Is there something out there ? inferring space from sensorimotor dependencies," *Neural Computation*, vol. 15, no. 9, 2003.

[6] D. Philipona, J. K. O'Regan, J.-P. Nadal, and O. J.-M. Coenen, "Perception of the structure of the physical world using unknown sensors and effectors," *Advances in Neural Information Processing Systems*, vol. 15, 2004.

[7] Y. Choe and S. Bhamidipati, "Autonomous acquisition of the meaning of sensory states through sensory-invariance driven action," in *Biologically Inspired Approaches to Advanced Information Technology, Lecture Notes in Computer Science*, A. J. Ijspeert, M. Murata, and N. Wakamiya, Eds., vol. 3141. Springer, 2004, pp. 176–188.

[8] Y. Choe and N. Smith, "Motion-based autonomous grounding: Inferring external world properties from internal sensory states alone," in *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006, pp. 936–941.

[9] R. Pfeifer and C. Scheier, "Sensory-motor coordination: the metaphor and beyond," *Robotics and Autonomous Systems. Special Issue on "Practice and future of autonomous agents"*, vol. 20, pp. 157–178, 1997.

[10] ——, *Understanding intelligence*. The MIT Press, 1999.

[11] S. Nolfi, "Using emergent modularity to develop control systems for mobile robots," *Adaptative Behavior*, vol. 5, no. 3-4, p. 343:364, 1997.

[12] R. Téllez, C. Angulo, and D. Pardo, "Evolving the walking behaviour of a 12 dof quadruped using a distributed neural architecture," in *Proceedings of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, 2006.

[13] R. Téllez and C. Angulo, "Progressive design of neural controllers through staged evolution," in *Technical report for the Technical University of Catalonia (U.P.C.)*, 2007.

[14] R. Téllez, C. Angulo, and D. Pardo, "Highly modular architecture for the general control of autonomous robots," in *Proceedings of the 8th International Work-Conference on Artificial Neural Networks*, 2005.

[15] R. A. Watson, G. S. Hornby, and J. B. Pollack, "Modeling building-block interdependency," in *Late Breaking Papers at the Genetic Programming 1998 Conference*, J. R. Koza, Ed. University of Wisconsin, Madison, Wisconsin, USA: Stanford University Bookstore, 22-25 July 1998.

[16] R. Watson, "Modular interdependency in complex dynamical systems," in *Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems*, 2002.

[17] F. Gomez and R. Miikkulainen, "Incremental evolution of complex general behavior," University of Texas, Tech. Rep. AI96-248, 1, 1996.

[18] F. Gómez and R. Miikkulainen, "Solving non-markovian control tasks with neuroevolution," in *Proceedings of the IJCAI99*, 1999.

[19] H. Yong and R. Miikkulainen, "Cooperative coevolution of multiagent systems," Department of computer sciences, University of Texas, Tech. Rep. AI01-287, 2001.

[20] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1,

no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/ars/SubscriberArea/Volume1/39-42.pdf
[21] S. Nolfi and D. Marocco, "Active perception: A sensorimotor account of object categorization," in *Proceedings of the VII International Conference on Simulation of Adaptive Behavior*, 2002.
[22] F. Linåker and L. Niklasson, "Time series segmentation using an adaptive resource allocating vector quantization network based on change detection," in *Proceedings of the International Joint Conference on Neural Networks*, 2000.

Cognitive Spatial Representations for Mobile Robots - Perspectives from a user study

Shrihari Vasudevan, Stefan Gächter and Roland Siegwart
Autonomous Systems Laboratory
Swiss Federal Institute of Technology Zürich
8092 Zürich, Switzerland
{shrihari.vasudevan,stefan.gachter,r.siegwart}@ieee.org

***Abstract*—**
Robots are rapidly evolving from factory work-horses to robot-companions. The future of robots, as our companions, is highly dependent on their ability to understand, interpret and represent the environment in an efficient and consistent fashion, in a way that is comprehensible to humans. The work presented here is oriented in this direction. It suggests a hierarchical concept oriented representation of space that is based on objects. This work attempts to provide a "cognitive" validation to the proposed representation and also looks into ways of enhancing it. This is done by means of an elaborate user study experiment. Analysis of the data obtained from the user study provides a human perspective to the robotics problem. This work also attempts to put forward a more generic methodology in order to develop such a representation, to be able to map the robots sensory information to increasingly abstract concepts that describe the semantics of the space the robot inhabits. The work itself is aimed at radically improving the degree of spatial awareness of state-of-the-art robot systems. Thus, the theme of the work is - representation for spatial cognition.

I. Introduction & Related Work

The state-of-the-art in mobile robotics use representations that are suited solely to the task of robot navigation. Further, these are not human compatible and fail to encode much or most of the semantics in the environment. This leaves them with little scope for use in more complex and interactive tasks. This is also the reason that the level of spatial awareness in current robot systems is quite modest. The focus of this work is to address these deficiencies. In an attempt to address these issues, a probabilistic object graph based representation of space was proposed in [1]. This work was a pure engineering exercise demonstrated on a robot platform. The work reported here attempts to address the problem from a human perspective.

Increasingly intelligent robots are tending to be more-and-more socially interactive. In the future, intelligence and the ability to meaningfully communicate will be critically important factors determining the compatibility and acceptability of robots in our homes. Most works in mobile robotics have until now restricted themselves to navigation related problems. Thus, few works evaluate their concepts in human centered experiments. A recent work which attempted to understand the acceptability of robots among people through a user study is done in [2]. This work was done on the sidelines of [3], which was a recent large scale demonstration of the remarkable growth of personal and service robotics. The representation proposed in this work promises to enable robots to not only perform navigation related tasks but also to be more spatially aware and human-compatible machines that could inhabit our homes alongside us. With the rapid increase in the importance of human robot interaction, the need for evaluating the work through human centered experiments was felt necessary. Further, it was felt that such experiments could contribute positively to the enhancement of the work itself. With this view, an elaborate user study was conducted to understand human perception and representation of spaces. This report is a detailed review of the salient aspects of the study.

The representation suggested here takes inspiration from the way we believe humans represent space and also the notion of a hierarchical representation of space. Ref. [4] suggests one such hierarchy for environment modeling. In [5], Kuipers put forward a *Spatial Semantic Hierarchy* which models space in layers comprising respectively of sensorimotor, view-based, place-related and metric information. Since the introduction of the term *Cognitive Map* in Tolman's seminal work [6], many research efforts have attempted to understand and conceptualize a cognitive map. The most relevant works include those of Kuipers [7] and Yeap [8]. The former viewed the cognitive map as having five different kinds of information (topological, metric, routes, fixed features and observations) each with its own representation. Yeap et al. in [8], review prior research on *early cognitive mapping* and classify representations as being space based or object based. The approach proposed here attempts to take the best of both worlds.

II. Approach

This work attempts to find answers to questions such as - (1) What is meant by "cognitive", when applied to a mobile robot from an engineering perspective? (2) How can a robot form a "cognitive" probabilistic representation of space? (3) How "cognitive" is the proposed approach and (4) How can a robot understand and reason about places? It does not attempt to propose a new theory of the mind. It proposes a human compatible representation of space for mobile robots and attempts to evaluate / enhance it through the user study presented. The proposed approach is shown in figure 1. The principle idea is that by adding concepts (created for instance using the functionality of the underlying elements) to the representation, semantics can be embedded in a purely navigation oriented map. The result can be understood as a concept oriented representation of space.

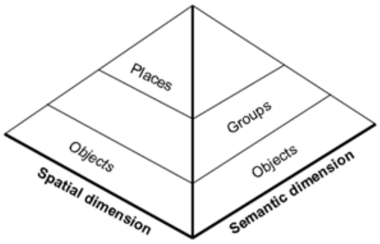


Fig. 1. The general approach - A robot uses the sensory information it perceives to identify high level features such as objects, doors etc. These objects are grouped into abstractions along two dimensions - spatial and semantic. Along the semantic dimension, objects are clustered into groups so as to capture the spatial semantics. Along the spatial dimension, places are formed as collections of groups of objects. Spatial abstractions are primarily perceptual or structural formations (occurrence of walls, doors etc.) whereas semantic or functional abstractions are primarily conceptual formations (similarity of purpose / functionality ; spatial arrangement). The representation is a single hierarchy composed of sensory information being mapped to increasingly abstract concepts.

The described approach has been partially implemented on a robot platform. The detailed approach is elicited in [1]. The perception system included methods for object recognition and door detection. The representation was probabilistic in order to account for the uncertainty and incompleteness of perception. Knowing the robots pose (using odometry) relative to a local reference, the detected objects and doors were identified in the local frame of reference. Using this information, a probabilistic graphical representation encoding the objects and the relative spatial information between them was formed as a local representation for the place. The local representations of different places were connected through the doors that connect them. Spatial Cognition was demonstrated through experiments on place classification and place recognition. More recently, promising results have also been obtained on the formation of concepts or groups; these will be reported very shortly.

III. The study

A. Overview

The survey comprised of a questionnaire posed to fifty-two people who were taken through a course in our premises wherein they were exposed to day-to-day objects and places. Due to the geographical location of this work (Lausanne, Switzerland), questions were posed in English or French, as the user preferred. The questions were based on the model presented in fig. 1. The survey was intended to be as unbiased as possible, without loosing the focus of the work itself. It was also attempted to make it as statistically representative as possible. Care was taken to ensure, to the extent possible, that age, gender, nationality and vocational background did not bias the survey in any way. However, expectedly, a majority of the survey takers were either Swiss or French. In the following subsections, various parts of the study are detailed, the results visualized and their implications analyzed. The graphs also provide some examples of replies that were obtained. Ref. [9] is a more explanatory version of this document with photographs of the various areas and objects used for the study as well as the questionnaire used.

B. Objects

1) Representation: Users were asked to imagine and describe how they represent typical objects such as a chair and a cup. The means of representation of a chair is shown in fig. 2. It was found that the structure was the dominant aspect of the representation. Few people actually used the functional description of a chair - an object on which people can sit. The material composition of the chair was a relatively more significant factor. A lot of finer detail was also obtained. These have been classified in fig. 3. Here it was found that the type of a chair and the level of comfort it offered were the relatively significant aspects of such descriptions.

With regards to the cup, most users seemed to use a structure based internal representation, as shown in fig. 4. Here also, the material composition of the cup was at an intermediate level of significance between the structure and the function. The finer details that accompanied the description, shown in figure 5, included more information on the shape and size of the cup.

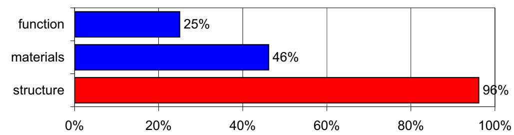


Fig. 2. Means of representation of a chair. Structure = { 4 legs, a seat, a back }, Materials = { wood, steel, plastic }, Function = { an object to sit on }

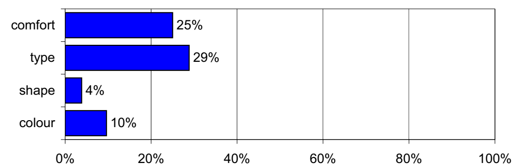


Fig. 3. Additional details used to represent a chair. Color = { brown, decorative patterns, black, dark } , Shape = { cubic shape, symmetric shape }, type = { kitchen chair, office chair } and comfort = { flexible, comfortable, rigid, cushioned }

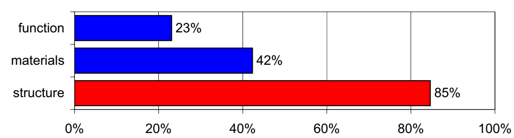


Fig. 4. Means of representation of a cup. Structure = { hollow object, container, handle }, Materials = { porcelain, ceramic, glass }, Function = { an object to contain or drink liquids }

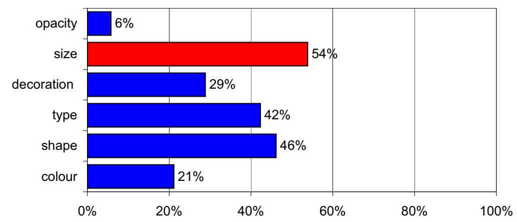


Fig. 5. Additional details used to represent a cup. Color = { white, plain color } , Shape = { cylindrical, round, oval, square }, type = { coffee cup, tea cup, espresso cup }, decoration = { patterns, text }, size = {various sizes, 0.5 l, 25-35 dl, 12 cm high and 67 cm in diameter }, opacity = { opaque }

2) Description of objects: In this part of the study, users were asked to observe and describe three objects - a traditional / simple chair, an office chair and a cup. The obtained descriptions were categorized as before. The means of description of the three objects are respectively depicted in figures 6, 8 and 10. The finer details of the description were also categorized and are depicted in figures 7, 9 and 11 respectively.

In the case of the office chair, the structure was the most important element describing it, followed by the type and the material composition of it. The finer detail obtained were primarily on the color and the comfort level that the chair offered. In comparison to this, while the structure of simple chairs was indeed the most important element, the significance between the remaining two factors was reversed. Also, the finer detail reflected more on the condition and the comfort level offered by the chair than its color. This is explainable since the traditional / simple chair used in this experiment is not particularly colorful or artistic, further it can be used in many different contexts and hence its *type* is indistinctive. The cup description also saw the greater significance of the structure over the type and the material composition. The finer detail of the cup was mostly concentrated on the decoration on the cup, its condition and size.

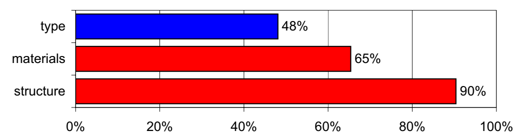


Fig. 6. Means of description of a simple chair. Structure = { 4 legs, a seat and a back}, Materials = { wood, steel, metal }, Type = { kitchen chair , school chair}

3) Object Recognition: Within the framework of these experiments, people were also queried on how they recognized typical objects. Their response was studied. Figure 12 shows the results of categorizing their answers. It was evident that most people used structural elements to identify objects.

4) Object arrangement: People were asked to describe a given scene (cupboard with objects above it and on its

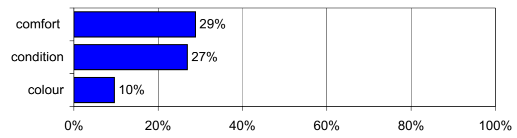


Fig. 7. Finer details in the description of a simple chair. Color = { brown }, Condition = { excessively used, dirty, slightly old }, comfort = { rigid, uncomfortable }

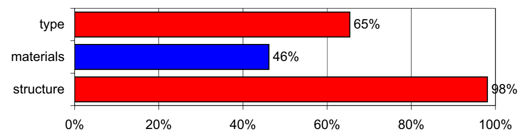


Fig. 8. Means of description of an office chair. Structure = { seat, back, axis with 5 wheels that roll }, Materials = { plastic, steel }, Type = { office chair, arm chair }

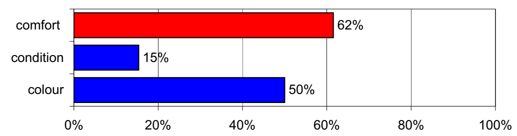


Fig. 9. Finer details in the description of a office chair. Color = { green, dark, colorful, decorative patterns on the cushion }, condition = { old, clean, nice }, comfort = { comfortable, ergonomic, adjustable }

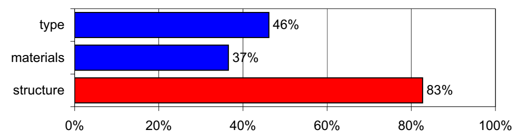


Fig. 10. Means of description of a cup. Structure = { cylinder with open top, hollow object, handle }, materials = { ceramic, porcelain }, type = { coffee cup, tea cup }

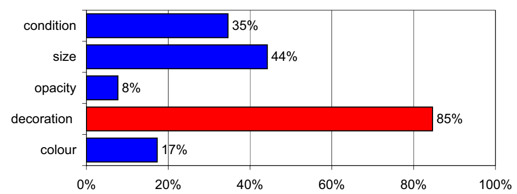


Fig. 11. Finer details in the description of a cup. Color = { white }, decoration = { picture of a dog and cat, gray background }, opacity = { not transparent }, size = { big, normal sized, 20 cl, 10-12 cm high }, condition = { used, stained, dirty, good condition }

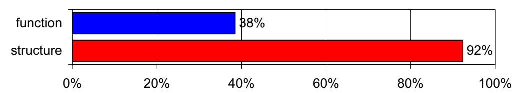


Fig. 12. Means of object recognition. Structure refers to the physical elements that make up the object whereas function refers to the more semantic aspects - functionalities of the object.

sides) in the refreshment room. Their response was studied. As shown in figure 13, people generally preferred to describe the scene from one end to the other (left to right or right to left) or in relation to a centralized object.

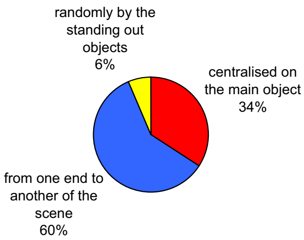


Fig. 13. Means of object arrangements in space

5) Analysis: The following summarizes the important observations made in the part of the study conducted above.

- The structure was the most prominent representative element for objects. This was followed by the material composition of the object.
- Descriptions of objects were also typically dominated by the structural information. However, the *type* of an object was found to be very relevant in this context. Similar but not the same as the *functionality* of an object, the *type* referred to the typical scenario in which the object was used.
- In both representation as well as description of objects, a lot of extra details were obtained, this could serve to enrich the proposed representation.
- Structure seemed to be the most important element in recognizing objects.
- Most people described the spatial configuration of a set of objects in an end-to-end fashion or less significantly, with respect to the central object.

C. Categorization

This part of the study aimed at understanding how people cluster and categorize space. The hypothesis under consideration was that people form explicit and implicit clusters of objects. This part of the study was aimed at understanding if the hypothesis is true, what were the basis of such clusters etc. Users were queried in different sized and featured environments about the existence of such clusters.

1) The Entrance Hall: People were taken first to an entrance hall of a building. As is usually the case, this was equipped with sofas, a telephone booth, some plants and some other tables and chairs. People were first asked to identify different zones [1] they observed and were then asked to justify their decision. The various zones that were formed in the entrance hall and the reasons for which they were formed are depicted in figures 14 and 15 respectively. Zones were mostly formed using groupings of objects and also due to the boundary elements. Typical zones identified as a result of grouping objects include a waiting zone (sofas, coffee table, plant etc.), phone zone (phone, chair, plant etc.), meeting zone(table and four chairs) and so on. Areas such as those near the entrance, in the corridor and near the stairs were identified as separate zones apparently due to the existence of boundary elements such as the walls, doors and the stairs. The meeting zone was sometimes identified by the objects composing it and at some other instances also included the boundary elements such as the window and the section of the stairs. Almost no-one used size as a metric to decide the existence of a zone. Further, it was found that objects were most often grouped due to the spatial arrangement they exhibited (for instance 4 chairs around a table) or due to the functionality they characterized or less significantly, due to the materials they were composed of.

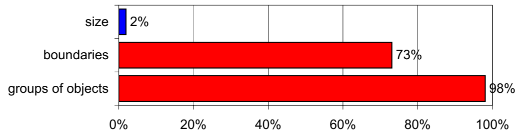


Fig. 14. Means of zone definition in entrance hall.

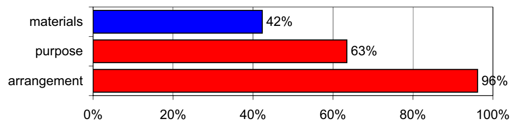


Fig. 15. Means of grouping objects in entrance hall

2) Laboratory / Office: People were next taken to a large laboratory-office. This room had 3 major areas within it - one part being a laboratory space with a lot of electronics and related lab-ware; the other part being meant for the people to work there and finally there was also a small round-table meeting area. As before, people were asked to identify if they saw any zones, if yes - which ones and why did they think that it was a zone? The results, when categorized appeared as shown in figure 16. In the case of every subject who took part in this study, the lab-office seemed to have 3 major zones within it. These zones were almost always identified by the objects lying around as the lab area (small electronic workshop) of the room looks significantly different to the office area (typical office). The objects clearly made out the zones. Many people also found the boundary elements within the room (partitions and an artificial wall made of cupboards) significant in that they separated the different zones. However, the general idea gathered from the study was that for most people, while the objects clearly grouped into 3 distinct regions within the room, the boundary elements were also useful but less significant and not absolutely necessary towards reaching this conclusion. Most often, the boundary elements were more supportive and less critical in the formation of the zones in this place.

3) Refreshment room: As in the previous case, people were taken to a refreshment room and asked to identify

[1]Refer section III-C.5 for details on the word *zone*

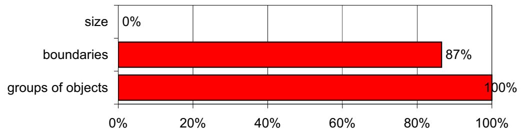


Fig. 16. Means of zone definition in a laboratory

zones within it. It is a relatively small room with a lot of diverse objects packed within it. The reasons for forming various zones in the refreshment room are illustrated in figure 17. The zones were almost always identified as a result of groupings of objects and both size and boundary elements were insignificant. The typical zones formed included a relaxing zone which comprised of objects like the sofas, the table and the surrounding plants, the kitchen zone which was the area having the coffee machine, kettle, microwave etc. and finally a book / storage / cupboard zone which housed a small library of books and archives of various technical journals and magazines. This was a case where object groupings were critical to the zone formation. Most people grouped objects due to their spatial arrangements and similarity of functionality / purpose. The rare exceptions included people who defined an entry zone based on the existence of a boundary element such as the door. Few people also viewed the whole room as a single place as they associated the three main functions of relaxing, eating / drinking and reading books as those that are common to a single place - i.e. they identified the 3 functionalities and felt that these were linked together and did not wish to identify them as separate zones.

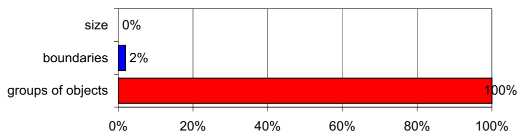


Fig. 17. Means of zone definition in a refreshment room

4) Office: Next, the users were taken to a typical office setting with three back-to-back/side tables a few cupboards and a table with a small experimental platform. As before, users were queried on the zones they could identify and the reasons why they did so. The results are depicted in figure 18, 19.

The office was a reasonably large room. Every single subject in the study identified the work place as a separate zone. This basically comprised of the three tables, chairs and work related objects that were on them. Many people also recognized the cupboards and shelf against the walls as a storage space. A few people seemed to perceive the existence of some experimental / robot hardware on a table as being a place for conducting experiments. A similar number of people perceived a separate zone just after the door and before the area containing the work related apparatus - this was termed as an entry space. A small number of people insisted on defining some sort of transition space between the entry space and the work place. What was particularly vindicating was that all the subjects formed zones through the grouping of objects and that the spatial arrangement of objects and their purpose were the two most contributing causes towards their being associated together to form the zone. A further experiment was also conducted in this place, the users were asked to explicitly cluster objects in the office. This resulted in users typically ending up reducing the grouping problem to that of a *classification of different kinds of objects* problem.

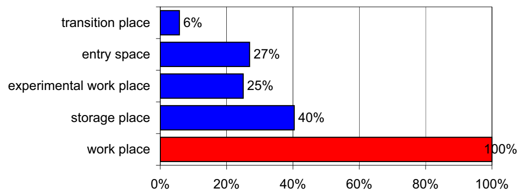


Fig. 18. Zone definition in an office

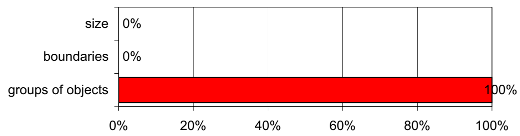


Fig. 19. Means of zone definition in an office

5) Analysis: The following conclusions could be drawn as a result of the experiments conducted in this subsection

- *On the word "zone"* - The word "zone" was used in order to avoid directly asking the user if they formed clusters and also taking into consideration local comprehensibility issues. The positive effect of framing the question this way was that user opinions were unbiased, yet they were not asked questions which they could not understand. The not-so-positive aspect of using this term was that people understood differently in different instances and replied with answers that had a mix of spatial and semantic abstractions. Users seemed to have understood a "zone" as being a standalone, separable unit, which was not quite what the survey designers wanted to convey. This made data analysis harder but nevertheless useful.
- *On the zones perceived* - People formed zones in the entrance hall - these were formed due to both object groupings and boundary elements such as the stairs that partition the entrance hall area. In the laboratory-office, it was observed that people were able to identify three separate zones even without partitions. The partitions of course, made this a more direct outcome. The refreshment room, being smaller in size, had only groupings of objects. The office also gave rise to zones - some were clear groupings of objects while some others such

as “entry space” and “transition space” had no obvious reason except the existence of a space that had a certain size and that clearly separated the user from a zone that had a certain clear and different semantics attached to it. Broadly, it was pointed out that there were two kinds of abstractions that were being produced - spatial and semantic (which are also termed as groups in this report). They are referred to here as semantic / functional abstractions (groups) and spatial abstractions (places). Semantic abstractions were almost always formed as a result of clustering objects - this clustering was typically the result of commonality in purpose and/or material composition, or specific spatial arrangements. Spatial abstractions were typically formed as a result of size and spatial elements such as doors, walls and partition elements. The exact effect of the former was clearly not well understood in this study and is being considered for future work. However, it was clear that intermediate level spatial abstractions (such as a small portion of a room partitioned off from the rest of the room, as in the case of the laboratory-office) were formed only when the size was significantly large. The presence of partitions led to the formation of a zone which is understood as an intermediate level spatial abstraction (or a mini-place of some sort within the place itself).

- *On the containment of semantic abstractions within spatial abstractions* - Given both spatial and semantic abstractions, there are two options on the hierarchy design - (1) to have both spatial and semantic abstractions at the same level of a hierarchy or (2) having a spatial abstraction contain a semantic one. In an office and in the refreshment room, there were quite a few people who, although identifying the existence of multiple functional areas (what are called semantic abstractions or groups here), did not want to split the room (a place) itself into those areas as they believed that these functional areas were an integral part of the same place. This fact seems to implicitly suggest the containment of semantic abstractions within the spatial abstractions. This seems intuitive, definitely valid for indoor environments and computationally suitable but probably could be more explicitly demonstrated. Thus, currently, as a design decision, spatial abstractions are chosen to contain semantic ones. However, proving/disproving this in a more explicit manner is something that should be addressed in future work.
- *On size dependence* - There is a clear dependence of size on the formation of zones. This was indicated in the lab-office and in the entrance hall. However this has not been explicitly addressed as this was a realization of the experiments themselves and the survey could not be modified at an intermediate stage. This would also be considered in the context of future work.

D. Places

The objective of this part of the study was to understand how people perceive different places. The working hypothesis here was that of an object based representation of space.

1) Representation: In this exercise, people were asked to imagine themselves in a place, such as a kitchen or an office. They were then asked to describe their perception in as much detail as they possibly could. The means of representing an office, a living room and a kitchen are respectively shown in figures 20, 21 and 22. Clearly objects and boundary elements formed the core of the replies obtained.

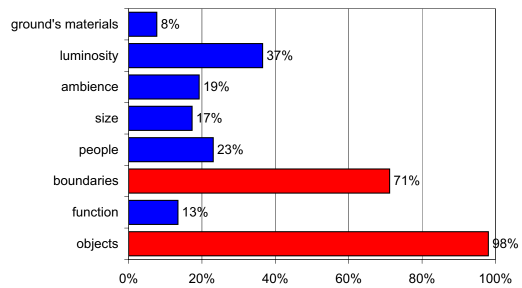


Fig. 20. Means of representation of an office. Objects = { desk, drawers, chair, books,...}, function = { place to work }, boundaries = { 4 walls, big window, door }, people = {several people sharing space, one person }, size = { not too large, 30 m^2 }, ambiance = { pleasant, active, sober }, luminosity = { natural light, artificial light, dimly lit, ‘strip’/tube lighting}

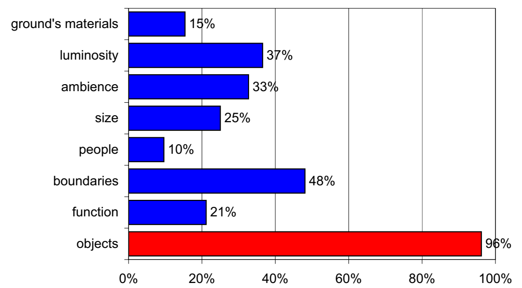


Fig. 21. Means of representation of a living room. Object = { sofa, armchair, coffee table, TV }, function = { place for rest, place to meet people, place to watch TV }, boundaries = { many windows, 2-3 doors leading to other rooms, high ceiling }, size = { big place, 40-50 m^2 }, ambiance = { calm, live, congenial, convivial }, luminosity = {natural light, very illuminated, big lamp for whole room }, ground materials = { carpet flooring }

2) Description: People were taken to different places and were asked to describe what they saw in as much detail as possible. The means of describing an office, a refreshment room and a laboratory-office are respectively shown in figures 23 24 and 25. In this case the objects and functionality of the place show significantly more importance than boundary elements.

3) Change of Place: The objective of this part of the study was to identify what leads to the formation of a place and how do humans sense that they are in a new place. People were taken from one place to another and queried as to when and why they believed that they were in a new place. The categorized results are depicted in figure 26. Clearly, boundary elements such as doors and walls and

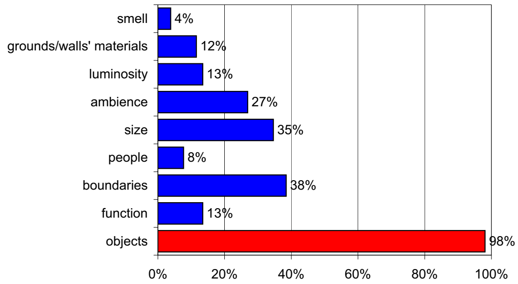


Fig. 22. Means of representation of a kitchen. Objects = { cooker, oven, fridge }, function = { place to eat, place to cook }, boundaries = { many windows, 2-3 doors leading to other rooms }, people = { family, kids }, size = { small, not very big, spacious }, ambiance = { sober, functional, clean }, luminosity = { bright }, ground materials = { tiled flooring }, smell = { food, good smell }

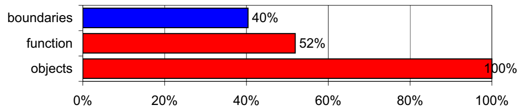


Fig. 23. Means of description of an office. Object = { desks, drawers, chairs, pens }, function = { place to work } and boundaries = { windows, closed space }

object arrangements constituted the most important criteria determining a change of place.

4) Analysis: The reason both representation as well as description details were sought from users was to maximize the data we have - both from a time-accumulated model (representation) of the place and an in-situ description. Objects were clearly the feature of choice when it comes to representing or describing places. Another significant element in this regard were boundary elements such as doors and walls. This seemed logical as the motivation for our approach is that objects provide the necessary semantics of the space while the boundary elements provide for the structure. However, boundary elements seemed relatively insignificant in the descriptions of places where the functionality was the principal component. Boundary elements such as doors and walls and object-arrangements turned out to be the most significant factors in determining a transition from one place to another. There were several other factors which were less significant but nevertheless worth consideration. Together, these could be understood as some sort of 'visibility' measure of a place.

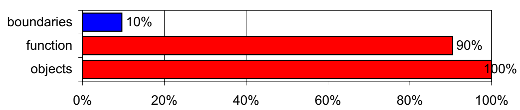


Fig. 24. Means of description of a refreshment room. Object = { sofa, armchair, table, shelves }, function = { place to relax, place for a coffee, place to read } and boundaries = { windows, door, walls }

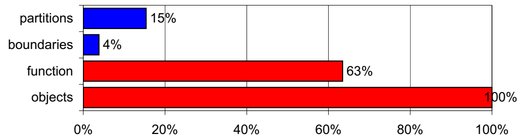


Fig. 25. Means of description of a laboratory-office. Objects = { workspaces(tables and chairs), wires, tools, oscilloscope, robots }, function = { place for theoretical and practical work }, partitions = { the shelves that separate the workshop like area from the office like area } and boundary elements refer to the typical boundary elements such as windows, door and walls

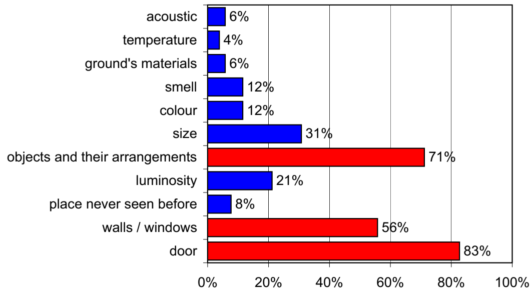


Fig. 26. Means of place formation / identifying a change of place. People cited a change in the objects and their spatial configuration, typical boundary elements such as doors and walls as a means of determining the occurrence of a new place. Other factors that were cited include a change in light intensity (luminosity), a sudden change in size / color / smell / ground materials/ level of sound and even temperature (places towards the interior of the building are significantly cooler than the areas near windows)

E. Hierarchical spatial representation

The objective of this part of the study was to somehow establish that a hierarchical representation may be a possible explanation to the way humans represent routes and space. People were asked to describe the route from the meeting point to the their current location (the same room for everyone). Their answers were categorized and are depicted in fig. 27.

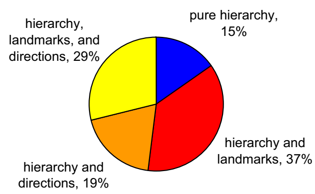


Fig. 27. Types of route encoding. A pure hierarchy represented a succession of path elements that simply jumped from one level of abstraction to another (for e.g. from building to floor to corridor to room), as mentioned before different people preferred to enrich their descriptions using landmarks, directions and even a combination of them.

1) Analysis: As observed from the graph above, every single subject gave some form of a reply that contained a hierarchical sequence of structural elements such as the floor - corridor - room sequence etc. However, people also made use of landmarks, directions and distance measures. The exact proportion of the combinations were also quantified and this seemed to suggest that a hierarchical sequence of structural elements with a combination of one / more landmarks was the most popular option.

IV. Discussions & Future work

The study brought out some interesting perspectives from the point of view of the overall approach. The most significant ones are listed below. Some aspects of the survey turned out to be well addressed while others could do with better treatment; there are still others which are too difficult to truly address and it is hard to find an appropriate way to glean or infer such information ordinarily.

- The word "cognitive", in the context of this approach, is more likely human-compatible and not necessarily human-like. Design decisions of the approach and results from the survey guarantee that the representation so formed is cognitive in that it is human compatible but are insufficient to estimate the similarity with the representation of the information in our brains. Future work will attempt drawing parallels and understanding the exact differences between the various schools of thought on the brain's cognitive map - as perceived by cognitive psychologists, neuroscientists and roboticists. Work would also be dedicated towards conducting more insightful user studies on the theme of this work.
- An object based representation is indeed useful for robots to develop a human compatible representation of space.
- Objects are grouped into groups or concepts - these are the semantic / functional abstractions in space. They are mostly formed by similarities in purpose, functionality and also by the relative spatial arrangements of objects.
- Places can be understood as spatial abstractions which are typically formed by bounding elements such as walls and doors.
- The survey brought out to a significant extent, the various properties, functionalities that may be relevant towards enhancing the representation being pursued.
- Typical groups (functional groupings of objects) that are formed by humans were also identified in the places where the users visited. This does give some ideas for other kinds of places too.
- Spatial abstractions contain semantic ones. This was indicated to a certain extent and subsequently taken as a design decision.
- A change of place was typically identified as a result of the occurrence of boundary elements such as doors or walls and also that of a significant change in the kinds of objects and relative spatial arrangements of objects.
- The structure of objects is critical to representing or describing them. Its material composition and type were also important. Scene descriptions were typically end-to-end or based on a central object.

The following issues warrant further research. Most have been attempted in this work. They produced results that were deemed insufficient (as discussed in earlier sections). Some are very hard to actually ask without biasing subjects. Some others were intermediate realizations of the study itself.

- Is space actually represented as a hierarchy? Are there spatial and semantic abstractions in our brains?
- Does the spatial abstraction contain the semantic one?
- The role of human activity. This issue is beyond the scope of the presented study. Both at the level of objects, how they are classified and at the level of functional spatial representation, this issue needs to be studied.
- In the context of object recognition, is the structure alone important? When and why does functionality come into play?

The representation proposed herewith, can enable a robot to develop a human-compatible representation of space and even a human-like conceptualization of space. It can equip robots with more than "just" navigational capabilities, make them much more spatially cognizant machines and yet ensure that they are still compatible and acceptable to us. This report sought a human perspective towards validating the approach and a feedback on how the representation could be enhanced. Both tasks have been successfully addressed. The study provides an empirical basis for certain facts that seem to be taken as obvious or concepts that are intuition inspired. Questions that are yet to be addressed or insufficiently addressed in this work were also identified for future work.

Acknowledgements

This work has been supported by the EU Integrated Project COGNIRON (The Cognitive Robot Companion), funded by Contract FP6-IST-002020. The work was conducted at the Swiss Federal Institute of Technology Lausanne (EPFL). The authors thank Marc Berger (EPFL) for his support.

References

[1] S. Vasudevan, S. Gächter, V. T. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots - an object based approach," *Elsevier Robotics and Autonomous Systems*, 2007, Accepted for publication in the special issue on "From sensors to human spatial concepts".

[2] K. O. Arras and D. Cerqui, "Do we want to share our lives and bodies with robots? A 2000-people survey," Autonomous Systems Lab (ASL), Swiss Federal Institute of Technology Lausanne (EPFL), Tech. Rep. 0605-001, June 2005.

[3] R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis, "Robox at Expo.02: A large-scale installation of personal robots," *Robotics and Automation Systems*, vol. 42, pp. 203–222, 2003.

[4] A. Martinelli, A. Tapus, K. O. Arras, and R. Siegwart, "Multi-resolution SLAM for real world navigation," in *11th International Symposium of Robotics Research*, Siena, Italy., 2003.

[5] B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, pp. 191–233, May 2000.

[6] E. C. Tolman, "Cognitive maps in rats and men," *Psychological Review*, vol. 55, pp. 189–208, 1948.

[7] B. J. Kuipers, *Spatial Orientation: Theory, Research, and Application*. New York: Plenum Press, 1983, ch. The cognitive map: Could it have been any other way?, pp. 345–359.

[8] W.-K. Yeap and M. E. Jefferies, "On early cognitive mapping," *Spatial Cognition and Computation*, vol. 2, no. 2, pp. 85–116, 2001.

[9] S. Vasudevan, S. Gächter, and R. Siegwart, "Cognitive Spatial Representations for Mobile Robots - Perspectives from a user study," Autonomous Systems Laboratory, Swiss Federal Institute of Technology Zurich (ETHZ), Tech. Rep. 165068-2006-01, 2006. [Online]. Available: http://www.asl.ethz.ch/people/vasudevs/personal

Posters

Knowledge Representation for Reconfigurable Automation Systems*

Ola Angelsmark[1], Jacek Malec[2], Klas Nilsson[2], Sławomir Nowaczyk[2] and Leonardo Prosperi

***Abstract*— We describe the work in progress on knowledge representation formalisms chosen for use in the European project *Skill-Based Inspection and Assembly for Reconfigurable Automation Systems*. The goal of SIARAS is to make agile manufacturing easier by creating an intelligent support system for reconfiguration and adaptation of assembly systems.**

I. SIARAS

SIARAS is an acronym of an EU-funded (FP6 - 017146) STREP-project with the general aim to support end users and engineers of manufacturing systems, including robotic ones, and to make production engineering easier (and thus cheaper) in most circumstances.

Its main goal is to build an intelligent system, named provisionally the *Skill Server*, capable of supporting automatic and semi-automatic reconfiguration of a manufacturing processes in response to changing requirements. The main issue during the design phase was to merge two, somewhat opposed, views on the reconfiguration process: the top-down, AI-based view and the bottom-up, engineering one.

A. Top-down AI approach

The top-down approach describes the problem of reconfiguration as a (re)planning problem: given a new task (usually expressed as a goal condition), possibly being a modification of the previous one, and given a set of skills available in the system, understood as a description of the operations that might be performed by the devices available to the user, find such a sequence of operations that would ensure that the task is correctly executed (find a plan that achieves the goal). It is assumed that the domain is modelled sufficiently well.

B. Bottom-up Reparametrisation Approach

In this approach the skill server is used only for reconfiguration of an existing, correct, properly modelled production system. The system is not expected to propose novel solutions, nor to search for alternative ways of implementing the process. In particular, one should expect a detailed description of the task: what is produced and how (i.e. what are the steps of the process). Moreover, for each step it should be clear how does it contribute to the goal. On the other side, available devices must be described in terms of operations they are able to perform (skills) and conditions under which they can operate. Skill Server needs to map task into skills and parametrise them appropriately.

* This work has been supported by the EU-project SIARAS (FP6 - 017146) http://www.siaras.org/.

[1] ABB, ola.angelsmark@se.abb.com

[2] Department of Computer Science, Lund University, Lund, Sweden, {jacek|klas|slawek}@cs.lth.se.

C. Finding the Golden Middle

It seems that the top-down AI approach is both computationally infeasible and impossible to model sufficiently well, while the bottom-up reparametrisation approach lacks generality and risks ending up as a database of *previously used* parameter settings for a number of devices in a number of scenarios. The main issue with this approach is guaranteeing scalability and extendibility to new domains or to new kinds of devices. There is a risk of limiting the approach to the previously considered cases and very similar ones only, thus precluding a more open-ended solution.

In order to make sure that we do not loose the larger perspective while we aim at restricting ourselves to a feasible problem, we imagine a layered approach, with reconfiguration level at the bottom and (re)planning level on top of it.

II. ONTOLOGY

We have decided to center knowledge representation around the concepts of devices (physical objects provided by their manufacturers) and skills (operations that can be performed). Task descriptions exist only during problem solving sessions, as dynamic structures, specific to a particular case. They can be seen as (arguably, quite complex) combinations of skills and parameters and therefore there is no need to have them explicit in the vocabulary.

The static part of the knowledge is represented in an ontology: a data structure storing all the necessary relations between the terms used. While ontologies are often used for classification purposes, in our case the classification is done when objects (skills or devices) are introduced in the structure. The main use of the ontology is to allow reasoning about skills matching particular tasks and about devices being suitable for particular operations, as well as to standardise the nomenclature used and the relationships of different concepts.

A carefully chosen set of representation primitives, together with a rather relatively ontology and a set of reasoning algorithms (available for complex formal systems such as description logics) allow us to keep the extension possibility open while focusing, in the beginning, on a concrete demonstrator case. As the project is done in cooperation with a German/Greek company INOS which provides system integration for automotive industry, we concentrate on a number of test cases from that domain.

We have chosen the open source tool Protégé for ontology creation and manipulation (together with reasoners such as Racer, Fact++ and Pellet), JGrafchart for task representation using Sequential Function Charts, and Python programming language for "gluing" the two together in the prototype.

Semantic-Knowledge-Based Execution Monitoring for Mobile Robots (Extended Abstract)

Abdelbaki Bouguerra and Lars Karlsson and Alessandro Saffiotti
Mobile Robotics Lab
Center for Applied Autonomous Sensor Systems
Örebro University, Örebro, Sweden 70182
http://www.aass.oru.se

Plan-based approaches have been a major trend in developing mobile robotic architectures capable of accomplishing complex tasks in non-structured environments. An important challenge that is faced by such architectures is how to carry out the execution of their plans so that tasks are achieved despite the presence of uncertainty and the dynamics of the real world. It has been recognized since the first days of autonomous mobile robots, that robust action execution requires monitoring the state of both the world and the robot to cope with contingencies that might occur at run-time.

Plan Execution monitoring approaches have generally focused on using the explicit effects of actions to derive expectations that are compared with what is really produced by the execution of the action. This supposedly means that the effects to monitor are directly observable. That is, of course, not always realistic in a complex environment where checking expectations is a complex process.

We propose to use more advanced forms of reasoning in execution monitoring, where semantic knowledge is used as a source of information to infer high-level implicit expectations related to the successful execution of plan actions. These expectations are derived and verified online, for each step of the plan that is being executed. As a concrete example of our approach, consider a mobile robot that is asked to deliver mail to the office of a certain person. As the robot enters the room asserted to be the office, it should expect to see at least a desk, a chair, and possibly a PC. These expectations are derived from the type of the room the robot is entering. If the robot is entering a kitchen instead, it should expect to see an oven, a sink,...etc.

By semantic knowledge we mean knowledge about objects, their classes and how they are related to each other. For instance, in an office environment, an office is a class whose individual instances (objects) denote rooms that have at least one desk and a chair; the entities desks and chairs are themselves defined as pieces of furniture ...etc.

Implicit expectations related to an object of a certain class refer to constraints over its properties and to relations to other objects (being in office we implicitly expect to see a desk).

Execution monitoring using semantic knowledge involves deriving implicit expectations, related to an object of interest, and checking them through a comparison process of the classification of what the robot has observed with what is expected i.e. the constraints that make the object of interest belong to its class.

This results in one of three outcomes:

- Consistent classification: this is the result when all the implicit expectations are verified i.e. the observed object is of the same type as the object of interest. Therefore, the monitoring module concludes that the action has been successfully executed.
- Inconsistent classification: this the result when one or more implicit expectations are observed to be violated. Therefore, the monitoring module concludes that execution of the action has failed.
- Unknown outcome: this outcome results when there is lack of information which makes implicit expectations not known to hold nor to be violated. Consequently, more information is needed in order to conclude whether the execution of the corresponding action has succeeded or failed.

To validate our monitoring framework, we implemented an architecture that includes an AI planning system, a monitoring module, a perception module whose input comes from a vision system, and the knowledge representation and reasoning system LOOM .

Experiments have been performed in a lab environment using a Magellan Pro mobile robot. Placing simple objects to simulate pieces of furniture to simulate a house environment. The full paper for this abstract is published in the proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA'07) held in Roma, Italy.

Failure Recovery in Robotics based on Acquired Semantic Information

Luís SEABRA LOPES, *Member, IEEE*

Abstract – **In complex task domains, robots need to reason about the tasks and the environment in order to make decisions. Reasoning should be based on semantic information acquired by the robot through experience as well as through interaction with users and other robots. This work focuses on the use of acquired semantic information for failure recovery in robotics. In the proposed approach, the basic principles that explain the success of a failure recovery strategy are extracted based on several deductive as well as inductive transformations. In recovery planning based on these learned principles, the inverse transformations are applied. The proposed approach is theoretically and empirically evaluated.**

I. Introduction

Service robots are expected to work in unstructured environments and to act, within certain limits, independently. Robots are also a central component in flexible manufacturing and assembly systems (FMS/FAS). In this context, the keyword *flexibility* is generally understood as *the ability to cope with change*. This is a particularly important topic, especially in what concerns failure detection, diagnosis and recovery, a problem that is far from receiving from the robotics community the due attention.

My work in recent years has been concerned with the development of robot architectures that support reasoning and learning at the task level. The long-term goal is to build robots that can be instructed in the domain of concepts of the human user and, preferably, in natural language (Fig. 1). Work towards this goal has been pursuing two lines of research: 1) using classical knowledge representation and reasoning for decision-making and dialog management; 2) using perception, user feedback and learning capabilities for grounding representations and language.

In this work, I focus on failure recovery, taking the common assumption in AI planning, namely that the world normally changes as described in action models. In case of failure, this work also assumes that failure analysis capabilities will enable to update the world model and start the recovery planning activity. The work finally assumes that the manipulated representations are, in some suitable way, grounded. As mentioned, grounding is one of my research interests, however, grounding research is not yet mature to support complex semantic representations and processing capabilities such as those described in this paper.

L. Seabra Lopes is with the Department of Electronics, Telecommunications and Informatics, Universidade de Aveiro, Campus de Santiago, P-3810-193 Aveiro, Portugal (e-mail: lsl@det.ua.pt).

II. Brief Description

This proposed learning/planning approach involves the recognition of the basic principles underlying solutions to concrete cases and the application of those principles to new situations. A concrete case is, in this work, a failure recovery episode, represented as a tuple <*OT*,*FT*,*RP*>, where *OT* is the *failed operation*, *FT* is the *failure instance* and *RP* is a plan (i.e. a sequence of operations) used to recover from the failure (*recovery plan*). The basic principles that explain the success of a failure recovery plan are extracted based on several deductive as well as inductive transformations, namely deductive generalization, abstraction, feature extraction and clustering of repeated plan patterns. The final result is a failure recovery schema, represented as tuple <*OT*,*FT*,*RSK*>, where:

- *OT* is the failed operation template.
- *FT* is the failure category template.
- *RSK* is the recovery plan skeleton, consisting of a sequence of abstract steps, where each step is a pair <*Action*,*Features*>, *Action* is either an abstract operator or a sequence of abstract operators and *Features* is a list of features of the objects manipulated by *Action*.

Therefore, from a single episode, a very expressive semantic structure is generated covering a wide range of distinct (although related) situations.

Failure recovery planning is guided by the recovery plan skeleton that is retrieved from memory, using the failure operation and failure instance templates as indexing key.

Formal analysis of the planning algorithm shows that, if extracted features (actually semantic relations) are enough and the situation is structurally similar, the search complexity of failure recovery planning will be linear in the size of the solution. Empirical tests carried out in a robotized assembly scenario based on the Cranfield Benchmark point to the same conclusion. For instance, a single failure recovery schema was used to generate solutions for five problems. These problems were quite different, the solutions ranging from 15 to 40 operations, but in all of them the effective branching factor of the search tree was close to 1.0 (ranging from 1.3 for the smallest problem to 1.1 for the largest problem).

An Experiment in Semantic Correction of Sensor Data

Stefan Stiene and Andreas Nüchter and Kai Lingemann and Joachim Hertzberg

Common wisdom has it that all knowledge has to go through the senses first. While this is sort of true, it is only part of the story. The other direction does also make sense: Expectation matters for perception. In semantic robot mapping, the two directions need to meet.

In prior work [1], [2], we have developed the technology for acquiring 3D geometry maps in 6DOF on a mobile robot, for interpreting data in terms of building structures (floor, walls, ceiling) and for detecting objects in the geometry data. Part of interpreting is to process the data using, e.g., matching and filtering algorithms. All these algorithms, however, were local and "syntactic" in the sense that the laser scanner data were massaged and squeezed out as good as possible, but there was no model-based feed-back from prior findings to subsequent hypotheses. There was no explicit expectation about what might be perceived.

Without a semantic model, errors in the sensor data could only be corrected locally in the sense of outlier rejection and the like. Model-based perception would allow furthermore to complete the data (I know the wall continues behind the bookcase, although I have never seen it) and to correct illusions (I can tell the image of a robot from a robot if the image is hanging high on the wall). This has been way beyond our previous approaches.

We describe here a first small step into the direction of model-based sensor data correction. It was motivated by a systematic error of our 3D laser scanner equipment, which, due to poor calibration of the pitch control servo, tends to map a ground plane to a slightly bent surface.

3D mapping of environments consists of several steps to be executed, namely 3D scan acquisition, range image registration, and global relaxation. Since every step may potentially introduce errors, we are using semantic constraints to reduce the errors in all steps. Scans are acquired by our robot in a nodding fashion of the 3D laser range finder. The controlled pitch rotation can only be performed with limited accuracy, so a horizontal plane scanned by the laser may not be perfectly horizontally adjusted in the measured data. The key idea of our horizontal scan justification is to extract scanned planes, i.e., the floor plane and the ceiling plane in the scanned data, and to readjust the 3D scan using these horizonal information, according to the following scheme:

a) Point Labeling: Using the algorithm of [3], all scan points are labeled as floor, ceiling, or object points, based on a local geometric criterion wrt. their neighbor points.

The authors are with the Knowledge Systems Research Group of the Institute of Computer Science, University of Osnabrück, Germany. {stiene, nuechter, lingemann, hertzberg}@informatik.uni-osnabrueck.de

Fig. 1. Left: Map, top view. Right, top: The unconstrained 3D mapping shows a banana-shaped form. Right, bottom: The horizontal justification and the constrained mapping lead to qualitatively correct maps.

b) Bottom Plane Extraction: For an estimate of a 3D plane, we use all points labeled as *floor*. First, an initial plane is estimated using three data points. Second, the plane is adjusted so that the mean point-to-plane distance is minimal for *all* floor points.

c) Scan Justification: Floor points are horizontal in an office building and all on one plane, unless a clear jump edge is measured. The extracted 3D floor plane is used to rotate the 3D scan, such that the estimated plane – and therewith the 3D scan – is horizontal, i.e., parallel to the ground plane of the first scan, which defines the coordinate system. Hereby, pitch and roll errors are corrected.

Preliminary experiments were carried out in an indoor office environment. 33 scans, containing 88000 3D data points each, have been acquired. Fig. 1 shows the qualitative result of the constraint mapping. For quantitative results, we compared several distance measurements both in the map and in reality, using a high precision distancemeter (Leica DISTO). The accuracy of the constrained map differs only by several centimeters from ground truth, with a mean error of 2.11%, compared to 3.19% in the non-constrained case.

In the end, data-driven interpretation and model-driven data correction would have to come together. Fusing them is obviously a hen-and-egg problem. In probabilistic robotics, an EM-type approach might be suitable. For the time being, we would opt for a more flexible way of integrating the different knowledge sources that are relevant for the overall process, favoring a classical blackboard architecture. In addition, floor classification comes in handy for object segmentation and interpretation as well as path planning.

References

[1] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM with Approximate Data Associoation. In *Proc. ICAR*, 2005.

[2] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor en vironments. *Journal RAS*, 45(3 – 4), 2003.

[3] O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proc. ICRA*, 2004.