In Proceedings of workshop on Ambient Intelligence AI*IA-03, Pisa, Italy, 2003

RoboCare: an Integrated Robotic System for the Domestic Care of the Elderly

S. Bahadori[1], A. Cesta[2], G. Grisetti[1], L. Iocchi[1], R. Leone[1], D. Nardi[1], A. Oddi[2], F. Pecora[2], and R. Rasconi[2]

[1] Dipartimento di Informatica e Sistemistica
University of Rome "La Sapienza"
Via Salaria 113, I-00198 Rome, Italy
{bahadori, grisetti, iocchi, leone, nardi}@dis.uniroma1.it

[2] Planning and Scheduling Team
Institute for Cognitive Science and Technology
Italian National Research Council
Viale Marx 15, I-00137 Rome, Italy
{cesta, oddi, pecora, rasconi}@ip.rm.cnr.it

Abstract. This article reports the state of advancement of the RoboCare project, which was launched in December last year to address the problem of providing assistance to elderly people using a combination of software, robots, intelligent sensors and humans. It addresses the creation of a multi-agent environment in which all these actors cooperate synergistically in order to provide user services. This paper details two aspects of the system in the making, namely the centralized, service-oriented supervision infrastructure, called Active Supervision Framework (ASF), and the robotic components in use in the present stage of system development. In particular, we give an overview of the integrated planning and scheduling, execution monitoring and diagnosis services offered by the ASF, and on the domestic testbed environment which has been realized in our labs for on-site testing of the integrated system.

1 Introduction

Many elderly people and others with severe physical or mental disabilities need constant care and support in all aspects of daily life, such as decision-making, reminding them of tasks or events and warning of dangers. Such intensive support is expensive to offer using human helpers.

The RoboCare project (http://robocare.ip.rm.cnr.it) was launched in Italy in December 2002 to address this problem using a combination of software, robots, intelligent sensors [1] and humans to provide powerful and wide-ranging assistance and guidance services such as health care for the elderly.

RoboCare shares some aspects with other high-tech projects for the assistance of elderly people, such as the Assisted Cognition Project [23] or the Pearl Mobile Assistant [31], but for the first time addresses the creation of a multi-agent environment.

The philosophy underlying this application of cognitive technology is to enhance the quality of care for the elderly by deploying a range of autonomous expert systems which co-operate to provide services.

For instance, Alzheimer patients would benefit from a system which reminds them about certain sequences of actions to be performed such as switching off the gas after turning off the stove or wearing glasses before wandering out of the bedroom. In its simplest form, such a system could be made up of a team of agents like a gas monitor and a mobile robot capable of advising the assisted person. All the actors in the system would be capable of carrying out individual reasoning, but would also need to collectively reason about the situations which can occur.

Creating such tightly coupled intelligent systems presents an important challenge for the artificial intelligence and robotics communities, requiring the development of intelligent agents which are capable of complex symbolic reasoning tasks and high levels of interaction with humans.

It is true that researchers have already been successful in creating teams of robots [19] which are capable of co-operating in tasks such as foraging, the exploration of unknown environments and simplified forms of soccer-playing. And the artificial intelligence community has made significant advances in recent years in the study of planning and scheduling, and has developed mature technology for automated problem solving. However, the realization of a complete system to assist the elderly in a real-world environment such as a health-care institution or a home requires tight integration of these technologies. Not only: a central issue in developing support systems for the care of the impaired is *robustness*. From the software point of view, this means that the supervising entity must be capable of adapting to unexpected or unforeseeable perturbations in the nominal behavior of the system. On the other hand, the robotic components must be endowed with flexible behavioral characteristics, in order to successfully and safely react in any contingency.

The system must provide more than loose co-ordination between its components. It must provide a complete supervision framework [29] which implements a solid infrastructure [2] and maintains a global view of the system and provides control functions for human operators. And, as caring for elderly people is often all about routine, such as scheduling walks and daily social events, the system should be capable of planning and scheduling such needs, as well as interpreting them when they occur.

The RoboCare project also examines the issues surrounding user-acceptance of technology helper systems, looking for example at the psychological impact of robotic care helpers on elderly people. Since the level of social interaction which can be obtained between robotic agents and human beings is related to familiarity, we are also investigating the use of Sony's AIBO canine robots (`http://www.aibo.com`) for monitoring tasks.

Overall, the issues which need to be addressed in the development of complex systems for the care of the elderly include communication, knowledge representation, human-machine interaction, learning, collective and individual symbolic

reasoning – to name but a few. What follows is a detailed account of the work carried out by our Research Units in the context of RoboCare.

This article is organized as follows: in Section 2 we will introduce the basic framework we are building to address the system supervisory functionalities, presenting the cognitive services a preliminary version of the framework delivers. We then focus on the integrated planning and scheduling service, the first of the fully implemented services with which the framework will be equipped. In Section 3 we illustrate the robotic components which are being developed for the RoboCare system, also detailing the structure of one of the environments in which the system is to be deployed. We end with a concluding section, in which we outline some medium and long term research directions in the context of the RoboCare project.

2 The Active Supervision Framework

In the context of the RoboCare project, we are concerned with the development of a control infrastructure, also referred to as *Active Supervision Framework* (ASF), which manages and supervises human agents, robotic systems, sensors and domotic components in an environment which supports a high level of human-machine interaction. All the actors of the system cooperate in order to provide services for human assistance.

Two possible environments have been identified as target applications for the ASF. On one hand, a domestic environment, in which the goal of the system is to assist elderly people in their daily life, predicting potential hazardous situations and delivering both cognitive and physical support to the assisted. On the other hand, we have focused also on a health-care institution scenario. The role of the ASF in this context would be predominantly that of enacting a predefined workflow of activities, i.e. compiling an efficient schedule of tasks to be carried out.

In this work we do not focus on the specific tasks implemented for the two target scenarios. Even if the two environments pose different challenges and have different requirements, it should be clear that in many aspects the types of functionalities which the system must implement are overlapping. Basically, the role of the ASF can be rephrased more simply as that of "deciding and monitoring courses of actions for the whole environment".

The scope of this section is to introduce the reader to some issues and initial results that we are starting to encounter in our research. After showing the basic architecture of the ASF, we will focus on the planning and scheduling service we are implementing in the ASF, through which it is possible to achieve highly coordinated workflows of activities. Also, the planning and scheduling module is being integrated with execution monitoring and diagnosis functionalities, thanks to which it is possible to close the loop around the integrated reasoning system.

2.1 The Supervisor Agent

The ASF is conceived as a supervisory agent which is responsible for exporting deductive functionalities, or *cognitive services*, for the underlying multi-agent system. It is a centralized module, which is endowed with a *global view* of the system and its state (see Figure 1).

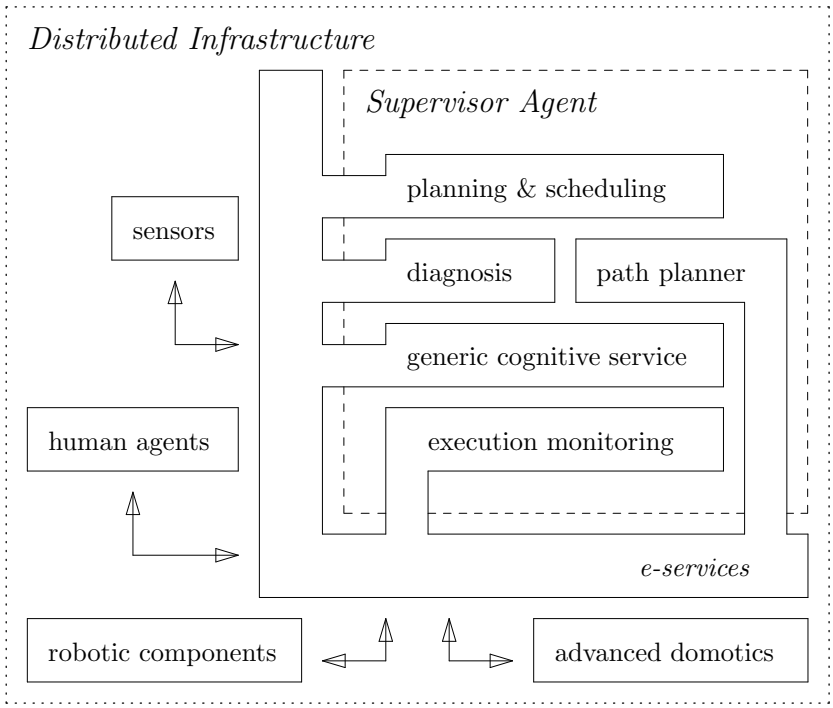

Fig. 1. The Active Supervision Framework, a service providing infrastructure for the RoboCare project.

The (necessarily) distributed nature of the application is achieved through the use of a distributed infrastructure. Having such a transparently distributable system makes it possible to separate logically-grouped functionalities on different platforms, thus enabling us to allocate computational resources more efficiently.

The overall function of the ASF is twofold. On one hand, the centralized framework is responsible for implementing active cooperation among multiple agents. This is one of the hottest topics in multi-agent systems today, namely because we must identify where to draw the line between centralized decision making and distributed team deliberation. On the other hand, the ASF is conceived as a supervising entity, through which human operators can control the

system at a high level, modifying the operational characteristics of the system and contributing to the execution of tasks in a mixed-initiative setting.

The functionalities of the ASF are made available to the actors of the system in the form of *e-services*. These services are invokable by all components of the system, although clearly most services are designed for either the robotic platforms or the human operators.

As a consequence of the above considerations, some of the functionalities the ASF offers are:

Path planning. The mobile robotic components of the system can resort to this service in order to obtain high-level paths to destinations which are not deducible locally due to the insufficiency of their sensory capabilities (e.g. an object which is reachable only by traveling through a set of interconnected rooms and corridors).

Integrated planning and scheduling. This service is responsible for determining workflows of activities in a goal-directed fashion. It implements a combined planning and scheduling reasoner, which delivers high-level plans for execution on behalf of the system components (humans, robots and so on). The integrated planner derives courses of actions starting from a symbolic representation of the environment and of the resources for execution, taking into account goals expressed in propositional form and accommodating time and resource constraints.

Execution monitoring. Human operators can rely on a set of tools to monitor the correct execution of predefined activity workflows, which are capable of predicting the effects of exogenous events in the schedule, such as activity delays, resource collapse and so on.

Diagnosis. Based upon the observation of the environment, this module reconstructs a model of the state of the executing components. In particular, the diagnostics module is capable of deriving the *reasons* for execution delays, failures and so on, and of determining the necessary adjustments in the high-level model of the environment used in the integrated planning and scheduling instances.

Generic cognitive services. In general, a functionality which satisfies any of the following requirements will be implemented as a module of the ASF: (1) computational issues inhibit its execution on local components (e.g. onboard memory and/or computational constraints of the robotic components); (2) global observability is required (e.g. knowledge concerning the dislocation of people or objects in the environment); (3) functionalities which are required by a variety of heterogeneous actors (e.g. consulting an agenda or remote-control tools)

2.2 Integrated Planning and Scheduling Service

Let us now focus on the integrated planning and scheduling service of the ASF, showing the assumptions we make at the planning stage that enable us to tackle

the multi-agent planning issue as a classical single-agent problem, an approach which induces a team of agents to cooperate in order to achieve a common goal.

The integration of planning and scheduling has already been dealt with, namely with software tools that can interpret a domain description language (DDL) that is capable of expressing time and logics, and is able to reason about both aspects of the domain. This approach is described in [11, 17, 22], just to name a few implementations. Others have dealt with an approach that separates the planning and scheduling phases, e.g., [32]. We are currently investigating the applicability of an approach similar to the latter, called *Naïve Component-Based Approach* (N-CBA).

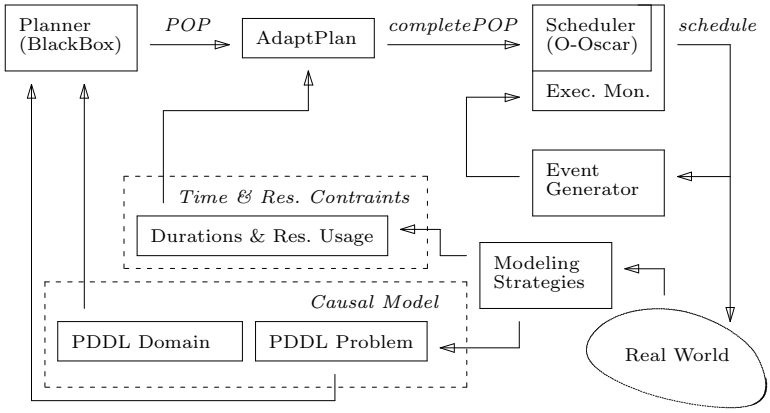

Fig. 2. The integrated planning and scheduling service, implemented according to the N-CBA.

The N-CBA consists of serializing a planner and a scheduler, as depicted in Figure 2. The term "Naïve" derives from the fact that this form of integration of planning and scheduling is very simple to achieve, and in certain aspects constitutes a theoretical starting point for more advanced forms of integration, as shown in [30].

A planner is given a problem specification along with a domain description *which is purified with respect to time- and resource-related constraints.* This part of the problem specification represents the *causal model* of the environment, and is reasoned upon by a partial-order planner. In turn, this reasoner produces a logically coherent, partially-ordered set of activities (Partially Ordered Plan – POP) which achieve the logical goal specified by the invoker of the service. The time and resource constraints are accommodated after the planning procedure has taken place through the plan adaptation procedure (AdaptPlan). This component is responsible for inserting the additional information concerning the duration and resource usage of the activities. A scheduler is then employed in order to obtain a completely instantiated solution of the initial planning problem.

A preliminary implementation of this integrated planning and scheduling reasoner built according to the N-CBA is capable of efficiently obtaining instances of activity worflows in a PDDL model of the health-care institution environment. Thanks to the de-coupling of the causal and time/resource-related reasoning, we are able to solve large problems with tens of executing entities and complex goal specifications. This implementation of the N-CBA makes use of established off-the-shelf components for the planning and scheduling modules. The choice of these components (or better, of the planning system) is strongly grounded on the theoretical analysis of the properties of the N-CBA made in [30].

The planning system which has been used is BlackBox, a Graphplan-based planner which combines the efficiency of planning graphs with the power of SAT solution extraction algorithms [27, 28]. This planner has done very well in recent planning competitions, and in the category of those planning paradigms which seem fit for planning and scheduling integration, it can certainly be considered to be one of the best choices. The scheduler which has been selected for use in this implementation of the N-CBA is O-Oscar [9], a versatile, general purpose CSP solver which implements the ISES algorithm [8].

The O-Oscar scheduler is also equipped with an *execution monitor* and an *event generator* [7]. In fact, the scheduling subsystem is capable of reacting to contingencies (produced by the event generator) such as task delay and resource collapse by adapting the completely instantiated plan *during execution*. This guarantees a minimal level of robustness in plan execution, given that the degree of determinism of such a complex system may be rather low.

The result is an architecture which is effectively capable of solving quite a number of problems, and is now being integrated into the ASF in the form of an e-service. It is not trivial to notice that this component based form of integration has some important advantages:

- the only necessary additional development is the adaptation procedure, which accommodates time- and resource-related constraints into the POP;
- assuming the separability of causal and time/resource-related problem specifications, the system performs quite well, especially with respect to plan quality;
- it is a general purpose tool, thanks to the PDDL planner input, and to the high expressivity of the scheduling problem specification [6, 3];
- the architectural de-coupling of the two reasoners allows for extensive human intervention when it comes to dealing with contingencies during plan execution. Indeed, a line of research we are following is *robust planning*, i.e. to analyze the properties of the causal plans (the POPs) in function of the degree of robustness of the schedules (completePOPs).

The performance details of this implementation of the N-CBA are outside the scope of this paper. These issues are extensively reported in [29], in which the N-CBA has been used to solve problems on a multi-agent domain which models the health care institution scenario.

The execution monitor has the task of assuring the completion of the plan while constantly checking its consistency. In a real working environment which

involves the presence of human beings, there are several events that may interfere with the correct execution of the plan, and each one of these events must be promptly counteracted [10].

Scheduled activities may be liable to unexpected delays (i.e. the administration of a medicine may be delayed due to a temporary unavailability of the patient); in other cases, an "unorthodox" behavior of the patient may cause some planned activity to last longer than expected, and finally, some required resources may suddenly turn out to be unavailable (i.e. unexpected loss of a robot or a specific sensor). Since the occurrence of any of these events may introduce inconsistency in the plan, a quick revision of the course of action is likely to be necessary. Such revision may involve some re-scheduling of the remaining activities, as well as a reformulation of the plan in the worst cases.

As we have seen, the integrated reasoner is capable of deriving high-level plans from a model which contemplates the causal and time/resource-related characteristics of the environment. Also, it is able to react to small contingencies by performing re-scheduling or adjusting the schedule which is being executed. But these elements of fault-tolerance do not really close the loop around the real world the system is to be deployed in. In fact, the next step is to furnish the system with diagnostics capabilities [3], which allow the reasoner to:

- *deduce* the state of execution of the plan;
- *change* the problem specifications in order to reflect the deduced state of the environment.

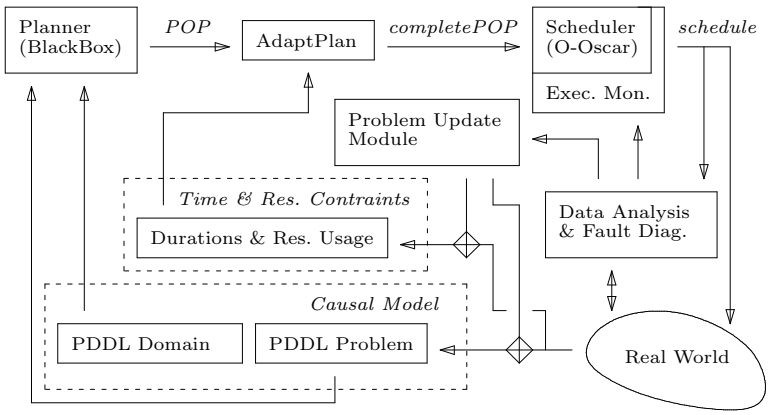

Fig. 3. The final deployment view of the integrated system, which implements the integrated planning and scheduling, execution monitoring and diagnosis services.

[3] The study and implementation of the data analysis and fault diagnosis module is a joint effort of the PST and another unit of the RoboCare project based at the University of Turin and lead by prof. Torasso.

As shown in Figure 3, the *data analysis and fault diagnosis module* replaces the event generator and integrates the system by providing a means to determine whether complete re-planning or re-scheduling is needed, and by automatically updating the problem specifications and constraints according to its deductions.

3 Robotic Components

A fundamental principle of the project is that the safety of the end users of the system must be of primary importance. Thus, we have decided to eliminate all the tasks that involve corporal care (i.e. cutting hair and nails, helping in body-wash and so on). According to this premise we focus on six achievable goals:

Reminder	("it is time for your pill")
Entertainment	("let's have a chess match" or "read me the news")
Transport	("get me a coffee from the coffee-machine")
Examination	("check the stove")
Search	("look for Mr. Smith")
Escort	("go for a walk with Mrs. Brown")

The order is not casual. The first item in the list is also the simplest (an electronic talking diary); The Transport task is relatively more difficult to achieve (all the common problems of mobile robotics are present), while the last item is also the most complex (in the Escort task the robot is like a partner in a walk that entertains the human while it supervises that everything is all right, and is programmed to call for help in case of dangerous situations).

3.1 Robotic platforms

The hardware involved in the project is made up of four mobile robots and several fixed devices (described in 3.4). The robot team is composed of heterogeneous members which differ in cost and appearance.

- The first component is a steward-like robot, furnished with a touch-screen, a microphone and a speaker for the human-machine interface, a tray just below for light-weight transport (pills, a newspaper or a glass of water). It is equipped with an omni-directional camera (mainly for navigation) and a stereo camera (for distance evaluation). All the sensors are plugged into a powerful on-board notebook. The mobile base is the newest Activmedia Pioneer 3; it is a reliable unicycle, with high power autonomy and a good carrying capability of 23 kg. We have designed an add-on structure which has the function of protecting the notebook and keeping all its parts together.
- The second robot is similar to the first. The difference is the Pioneer 2 mobile base. Due to its limited payload it probably will not carry as many components as the Pioneer 3 based robot. It will have a camera for navigation,

Fig. 4. The team of robots

while we are investigating the feasibility of further component integration. We are planning to use a Tablet PC instead of the combination of a notebook and a touchscreen. This solution offers slower calculation but lighter hardware.

- The third member is a cheaper solution. We employ an ER1 robotic Kit from Evolution Robotics, which includes a ductile aluminum skeleton, motors, wheels, infra-red sensors and an arm-grip. Also in this case the on board computer is a notebook.
- A different kind of robot is the Sony-AIBO, whose locomotion is based on four legs instead of wheels. It will be involved mostly in the entertainment task as a entertainment dog.

We are investigating the possibility of designing a self-made platform: it will be like the steward-robot but for the mobile base we will use an holonomic system that has three degrees of freedom, thus it would be able to carry out complex maneuvers.

3.2 Robot Basic Capabilities

All the applications of mobile robotics have to deal with some common problems in order to make the robot able to move from one place to another and to recognize the components of the world:

Localization. Localization is the task of evaluating the robot's position in the environment, given a sensor history and the knowledge of a map. A wide range of localization methods have been studied, mostly based on an explicit representation of the robot position belief [16, 15] in which the position

estimate is iteratively refined as the robot moves and detects environmental landmarks. In the RoboCare environment several efficient and reliable solutions can be studied and implemented, like the ones proposed in [18]. Moreover, the ability to give structure to the environment allows for devising some external systems for automatically measuring the precision of a localizing robot, granting the possibility of performing accurate experiments and evaluations of the developed localization algorithms.

Safe Navigation. When robots have to move around in a real-world environment, they are required to navigate without colliding with things and human beings. We use tested techniques of computer vision (color segmentation, edge detection and heuristic algorithms) to identify obstacles in the environment and therefore to compute a free path that leads the robot to the desired position avoiding collisions. Several approaches in the literature address the problem of safe navigation [26] and particularly interesting for our application is safe navigation in dynamic unpredictable environments [33, 24, 5, 4, 25]. The approaches used to compute the free path can be based on local information (such as [5, 4, 25]), or on a complete search on the working space [33, 24]. Local methods ensure prompt reaction to unexpected changes in the configuration of the obstacles, but can get stuck in local minima, especially in highly cluttered environments. On the other hand, global methods are able to compute effective paths in cluttered environments, but are normally characterized by high computational requirements. We use a path-planning method [14] which integrates the information of obstacle dynamics in the path planning process, and is able to compute effective paths in real-time; the method has been tested in the RoboCup soccer competitions.

Human Body Detection and Tracking. Human body detection is one of the most sophisticated and challenging aspects of object detection. There are two types of issues in human body detection which have to be considered: on one hand, distinguishing the human body from other objects in an image frame or video stream, and on the other hand, presence detection and motion tracking. One way to do this is to divide the job in to three phases (Clustering, Classification, Modeling) [34]. As for human presence detection and localization, one of the known methods consists of using the dynamic contours and assigning them to each moving object in the scenario [12]. In order to ensure the safety of the people we fuse the information coming from some additional sensors (thermal, active position transmitter) with the images from the cameras.

3.3 The cognitive robotic architecture

Our experience in mobile robotics (we are in the RoboCup competition since 1998) is summarized in [13], where we describe a Robot Development Kit (RDK) for modular programming of multi platform mobile robots. We are currently using our framework on different kinds of mobile platforms and for different applications: Robotic soccer, RoboCup Rescue (an international project for building

search and rescue robots operating in post-catastrophic environments) and now RoboCare .

The toolkit we have realized includes middleware that implements all the basic requirements for the development of a typical robotic application, a set of modules implementing the basic task of the robot and a set of tools that are useful for developing, monitoring and debugging the entire application. We can see in Figure 3.3 the complete architecture. The lower layer RHI (Robot Hardware Interface) realizes the independence from the hardware (it is a robot-specific driver).

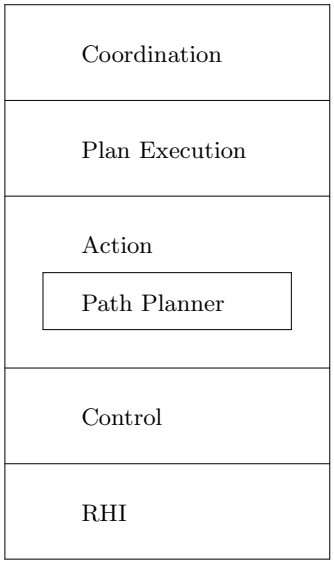

Fig. 5. The cognitive robotic architecture.

Our toolkit provides the main features required for the design and the implementation of a team of cognitive robots. They are:

- a framework for defining the robot's knowledge of the environment and its capabilities;
- a reasoning system able to derive plans (i.e. programs) that must be executed to achieve a given goal;
- a plan execution mechanism;
- a coordination module which is in charge of assigning sub tasks to each robot in the team in order to achieve a global team goal.

The cognitive robotic system, which will run on the heterogeneous robotic platforms we mentioned before, is based on a hybrid layered architecture, with two

levels: the Operative Level, in which the information is handled numerically, and the Deliberative Level, which relies on a symbolic representation of the robot's knowledge. In particular, at the Deliberative Level, the environment is described by a knowledge base containing both axioms and static facts (i.e. background knowledge) provided by the designer, as well as dynamic facts (i.e. symbolic information) corresponding to the data acquired through the robot's sensor during its mission.

Domain and Plan Representation. Our planning and reasoning framework [20, 21] relies on an epistemic representation of the world, which explicitly takes into account the robot's knowledge, rather than the true status of the world. The domain description includes the specification of the basic actions the robot can perform, which are defined in term of preconditions (conditions that must be verified for an action to be executed) and effects (conditions that are verified after the execution of the action). Our framework makes it possible to define two kinds of actions: ordinary actions - which produce changes in the world, and sensing actions - which produce changes in the robot's knowledge (i.e. in its mental state) and allows for sensing the value of a property that is not known. Plans are represented as transition graphs, whose nodes represent epistemic states and whose edges represent actions causing state transitions. The graph representing a plan has an initial node from which the execution starts and a set of final states in which the goal is satisfied. The execution of an ordinary action causes a deterministic state transition, while the execution of a sensing action causes a state transition depending on the sensing outcome. Moreover, since in the real world the execution of an action is not guaranteed to succeed, a set of recovery states are associated with each action.

Plan Generation and Execution. Also, we have defined a logic formalism and developed a plan generator. Given: i) the domain specification as a set of axioms; ii) the actions specification, in terms of preconditions and effects; iii) an initial state; iv) a goal, it generates a plan which obtains the goal from the initial state. Through the use of sensing actions, the plan which is generated can contain both `if` and `goto` constructs. It is also possible to manually edit plans using a graphic tool. A plan is executed by navigating the plan graph, starting from the initial node and terminating when a goal node is reached or when the plan fails. A plan failure can be caused by the failure of some of the execution conditions of the action, and will activate a plan recovery or a new plan selection depending on the presence of edges exiting from the recovery state.

Coordination. Coordination among robots is achieved by a coordination module that selects which sub-task must be accomplished by every robot. In our cognitive systems, each task is related to a goal to be achieved by a robot and thus to a plan to be executed. Tasks are assigned during robot operation by means of a distributed protocol that allows information about the current state

of each robot in the team to be shared, and provides the necessary autonomy in case of network failures. The state of each robot and its capabilities are evaluated in order to decide to which robot a given sub-task should be assigned and a broadcast agreement among all the robots is performed in order to effectively assign sub-tasks to robots, while avoiding interferences among them.

3.4 The RoboCare Domestic Environment

We have designed and realized an experimental domestic environment which reproduces a small flat composed of two rooms, a bedroom and a dining room with the kitchen. This will be the main theater for the mobile robots. In this environment they will assist an elderly person. Due to the fact that we want to simulate a residence for the elderly or a hospital, we will use the corridors of the institute to test the ability of the robots to work in large spaces. The robots are not the only components of the system. There are also a number of fixed devices that are useful to obtain precise information about the environment. We use fixed stereo-camera(s) for people and robot localization and tracking. The cameras are placed on the wall just under the ceiling, and face the ground in order to maximize the visible area. Another important device that we are planning to use is a grid of sonars mounted on the ceiling of the room that will be very useful in the localization of the robots. As we said in 3.2 these devices are responsible for measuring the accuracy of self-localization of the robots, thus enhancing the degree of reliability of the entire system.

4 Conclusions

In this article we have given an overview of the efforts of two research units involved in the RoboCare project. The work we are conducting is strongly based on our experience in the fields of planning and scheduling on one hand, and of multi-robot systems on the other. Nonetheless, we are strongly projected towards the unexplored area of integrating these two disciplines. The aim of RoboCare is to design and implement an intelligent infrastructure which makes use of autonomous robots and an array of state-of-the-art domotic components to be deployed in a real environment for the care of the elderly. The success of our ambitious project depends strongly on the synthesis of many disciplines. Hopefully, by the end of RoboCare we will have matured enough scientific and technological know-how to create a single, coherent outcome that can benefit some of the most vulnerable in our society.

Acknowledgments

This research is partially supported by MIUR (Italian Ministry of Education, University and Research) under project RoboCare (A Multi-Agent System with Intelligent Fixed and Mobile Robotic Components).

References

1. Aleotti, J., Caselli, S., and Reggiani, M. Development of a Sensory Data User Interface for a Service Robot. In *Proceedings of 1st Workshop on Advances in Service Robotics (ASER'03), March 13-15, Bardolino (VR), Italy.* (2003).
2. Amoretti, M., Bottazzi, S., Reggiani, M., and Caselli, S. Experience in Teleoperation System Design based on Real-Time CORBA. In *International Conference on Advanced Robotics, ICAR'03* (Coimbra, Portugal, 2003).
3. Bartusch, M., Mohring, R. H., and Radermacher, F. J. Scheduling Project Networks with Resource Constraints and Time Windows. *Annals of Operations Research 16* (1988), 201–240.
4. Borenstein, J. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on Volume: 7 Issue: 3* (1991), 278 –288.
5. Bruce, J., and Veloso, M. Real-time randomized path planning for robot navigation. In *In Proceedings of IROS-2002* (Switzerland, October 2002).
6. Brucker, P., Drexl, A., Mohring, R., Neumann, K., and Pesch, E. Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. *European Journal of Operations Research* (1998).
7. Cesta, A., Cortellessa, G., Oddi, A., Policella, N., and Susi, A. A Constraint-Based Architecture for Flexible Support to Activity Scheduling. In *Proceedings of 7th Congress of the Italian Association for Artificial Intelligence* (2001).
8. Cesta, A., Oddi, A., and Smith, S. A Constrained-Based Method for Project Scheduling with Time Windows. *Journal of Heuristics 8*, 1 (2002), 109–135.
9. Cesta, A., Oddi, A., and Susi, A. O-Oscar: A Flexible Object-Oriented Architecture for Schedule Management in Space Applications. In *Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-99)* (1999).
10. Cesta, A., and Rasconi, R. Execution Monitoring and Schedule Revision for O-Oscar: a Preliminary Report. Tech. rep., PST@ISTC-CNR, July 2003.
11. Currie, K., and Tate, A. O-Plan: The Open Planning Architecture. *Artificial Intelligence 52*, 1 (1991), 49–86.
12. Edelman. *Representation and recognition in Vision.* MIT Press, 1999.
13. Farinelli, A., Grisetti, G., and Iocchi, L. Design and implementation of modular software for programming mobile robots. *Software: Practice and experience* (2002).
14. Farinelli, A., and Iocchi, L. Planning trajectories in dynamic environments using a gradient method. In *Proceedings of RoboCup Symposium* (Padua, Italy, 2003).
15. Fox, D., Burgard, W., Dellaert, F., and Thrun, S. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceendings of the 16th National Conference on Artificial Intelligence (AAAI99)* (1999).
16. Fox, D., Burgard, W., and Thrun, S. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research 11* (1999), 391–427.
17. Ghallab, M., and Laruelle, H. Representation and Control in IxTeT, a Temporal Planner. In *Proceedings of the Second International Conference on AI Planning Systems (AIPS-94)* (1994).

18. Grisetti, G., Iocchi, L., and Nardi, D. Global hough localization for mobile robots in polygonal environments. In *Proceedings of International Conference on Robotics and Automation (ICRA02)* (Washington DC, USA, 2002).
19. Iocchi, L., Nardi, D., and Cesta, A. Analyzing Features of Multi-Robot Systems. An Initial Step for the RoboCare Project.
20. Iocchi, L., Nardi, D., and Rosati, R. Planning with sensing, concurrency, and exogenous events: logical framework and implementation. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000)* (2000), pp. 678–689.
21. Iocchi, L., Nardi, D., and Rosati, R. Strong cyclic planning with incomplete information and sensing. Tech. Rep. 16-03, DIS - Università di Roma La Sapienza, 2003.
22. Jonsson, A., Morris, P., Muscettola, N., Rajan, K., and Smith, B. Planning in Interplanetary Space: Theory and Practice. In *Proceedings of the Fifth Int. Conf. on Artificial Intelligence Planning and Scheduling (AIPS-00)* (2000).
23. Kautz, H., Fox, D., Etzioni, O., Borriello, G., and Arnstein, L. An Overview of the Assisted Cognition Project. In *Proceedings of the AAAI Workshop on Automation as Caregiver* (2002).
24. Konolige, K. A gradient method for realtime robot control. *AIROS* (2000).
25. Kreczmer, B. Application of parameter space discretization for local navigation among moving obstacles. *Robot Motion and Control, 1999. RoMoCo '99. Proceedings of the First Workshop on* (1999), 193 –198.
26. Latombe, J. C. *Robot Motion Planning.* Kluwer Academic Publisher, 1991.
27. McAllester, D., Selman, B., and Kautz, H. Evidence for Invariants in Local Search. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'97)* (Providence, Rhode Island, 1997), pp. 321–326.
28. Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L., and Malik, S. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)* (2001).
29. Pecora, F., and Cesta, A. Planning and Scheduling Ingredients for a Multi-Agent System. In *Proceedings of UK PLANSIG02 Workshop, Delft, The Netherlands* (2002).
30. Pecora, F., and Cesta, A. The Role of Different Solvers in Planning and Scheduling Integration. In *Proceedings of AI*IA-03, Pisa, Italy* (2003).
31. Pollack, M. Pearl: Mobile Robotic Assistant for the Elderly, 2002.
32. Srivastava, B., Kambhampati, S., and Do, M. Planning the project management way: Efficient planning by effective integration of causal and resource reasoning in RealPlan. *Artificial Intelligence 131*, 1-2 (2001), 73–134.
33. Thrun, S., Buken, A., Burgarg, W., Fox, D., Hennig, T. F. D., Hofmann, T., Krell, M., and Schmidt, T. Map learning and high-speed navigation in RHINO. In *AI-based Mobile Robots : Case studies of successful robot systems* (1998), MIT Press, Cambridge,MA, D. Kortenkamp, R.P. Bonasso, and R. R. Murphy editors.
34. Zhao, L. *Dressed Human Modeling, Detection and Parts localization.* PhD thesis, Carnegie Mellon University, 2001.