Information Sciences 179 (2009) 3973–3984

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Towards an architecture for semiautonomous robot telecontrol systems

Gracian Trivino [a,*], Luis Mengual [b], Albert van der Heide [a]

[a] European Centre for Soft Computing, Edificio Cientifico-Tecnologico, C/Gonzalo Gutierrez Quiros s/n, 33600 Mieres, Asturias, Spain
[b] Faculty of Computer Science, Polytechnic University of Madrid, Campus de Montegancedo, 28660 Madrid, Spain

ARTICLE INFO

Article history:
Received 6 September 2007
Received in revised form 16 June 2009
Accepted 6 August 2009

Keywords:
Robot telecontrol
Semiautonomous robot
System architecture
Human–robot interface

ABSTRACT

The design and development of a computational system to support robot–operator collaboration is a challenging task, not only because of the overall system complexity, but furthermore because of the involvement of different technical and scientific disciplines, namely, Software Engineering, Psychology and Artificial Intelligence, among others.

In our opinion the approach generally used to face this type of project is based on system architectures inherited from the development of autonomous robots and therefore fails to incorporate explicitly the role of the operator, i.e. these architectures lack a view that help the operator to see him/herself as an integral part of the system.

The goal of this paper is to provide a human-centered paradigm that makes it possible to create this kind of view of the system architecture. This architectural description includes the definition of the role of operator and autonomous behaviour of the robot, it identifies the shared knowledge, and it helps the operator to see the robot as an intentional being as himself/herself.

1. Introduction

The availability of a generic architecture is a great advantage when dealing with the project of designing a new computational system [3]. Usually this type of architecture is consolidated iteratively by subsequent implementations from which a general structural core starts to emerge. The research and development in autonomous robotics has produced different architectures which reflected the state of the technology at that time, e.g. deliberative architectures, subsumption architectures, hybrid architectures, behavior based architectures, etc. [24].

Over the years these types of architectures have found their way into the domain of tele-controlled robots. Early tele-controlled robots were directly operated by a human, usually in controlled environments with little uncertainty [15]. Current research has shifted the attention to semiautonomous robots, in other words, tele-operated robots with a certain autonomous functionality which are controlled by a human operator. These robots are designed for working in dynamic environments with some level of uncertainty [30]. Tele-operated robots are applied in many domains, for example, in space exploration [10], in potentially dangerous missions like anti-personnel mine clearance [12], in search and rescue missions after natural disasters or missions into collapsed man-made structures [26]. Other types of applications in which robots are used to assist the operator perform a task are, the night watchman task in museums [2], assistance in remote surgery [1], or the control of a wheelchair [31]. All these examples show forms of operator–robot cooperative control which demonstrates the potential of shared control which combines operator control with robot autonomy.

The design and development of a computational system to support robot–operator collaboration is a challenging task involving different technical and scientific areas, namely, Software Engineering, Psychology and Artificial Intelligence, among others [36].

* Corresponding author.
E-mail addresses: gracian.trivino@softcomputing.es (G. Trivino), lmengual@fi.upm.es (L. Mengual), albert.vdheide@softcomputing.es (A. van der Heide).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.08.007

Besides the habitual problems related to any medium sized project in Software Engineering, the design of this type of systems has a number of specific difficulties that have to be dealt with: finding a balance in robot autonomy versus operator control, dealing with the frequently occurring problem of an imperfect communication system [16], and most importantly, the creation of an understandable operator interface [11,35]. This last challenge is of critical importance when taking into account that most of the times the operator is responsible for making the final call and therefore should be conscious of what the robot is doing and what is happening at every moment in time.

Despite the diversity of semiautonomous systems we can identify a common architecture, i.e., a set of related main components which constitutes a generic structure for this family of systems [18]. In Software Engineering the system architecture is commonly organized in views [6]. These views are partial descriptions from particular perspectives with a focus on specific system characteristics, which hide details irrelevant in the current context.

Many people are involved during the development and exploitation of a system, for example, system developers, testers, quality control personnel, etc. These groups of people have different views of the system architecture and it would be useful to have a system architecture description adapted to the specific view of the user.

In our opinion, the general approach used when facing the project of developing tele-operated semiautonomous systems is based in architectures inherited from the development of autonomous robots and therefore fail to incorporate explicitly the role of the operator, i.e. these architectures lack a view that helps the operator to see him/herself as a part of the system.

The main contribution of this paper is to provide a paradigm called *CPA7L* that promotes a human-centered view of the architecture of tele-controlled semiautonomous robots. This architectural description includes and has special attention for the definition of the roles of operator and of the autonomous behaviour of the robot, it identifies the shared knowledge, and it helps the operator to view the robot as an intentional being as himself/herself.

This paradigm includes:

(a) A general scheme that is used to classify the different types of knowledge managed by the operator and the autonomous behaviour of the robot during their interaction.
(b) A hierarchical description of the system architecture that takes into account the special role of the operator.

CPA7L provides a generic view of the main components of this type of systems that can be used as a starting point for the development of new architectures, focussing on a human-centered approach that should deliver systems that are easier to design, develop and operate. Moreover the availability of a generic basic paradigm is essential when the goal is to design a new generation of semiautonomous robots with advanced capabilities as complex as, for example, learning by imitation [20] or interaction based on Natural Language [17]. Additionally the paradigm can be applied to Robotics in general with other objectives, namely, to create a human-centered perspective of already existing system architectures, to compare functionalities of different tele-operated robots, to make a didactic description of the system functioning, or to design and analyze the Human–Robot Interaction (HRI). (See examples of some of these uses in Section 4.)

In the rest of this paper we explain the paradigm *CPA7L*, creating a view of the system architecture focused on the elements the operator needs to know for understanding the system functioning. To illustrate this explanation we give a few details about the implementation of an HRI based on these ideas.

2. The Concepts–Procedures–Attitudes classification

In this section, we describe a classification of the three types of knowledge that the operator and the autonomous behavior of the robot manage during the mission.

The distinction between Concepts, Procedures and Attitudes (CPA) has its origin in Cognitive Psychology and appears in the Theory of Elaboration introduced by Reiigeluth and Merrill [29]. These researchers were investigating the contents of the learning process in the general context of Education. Some years later this classification was used as one of the theoretical bases of the Spanish educational system. In one of the seminal books definitions are provided of the key concepts of the theory [7]:

Concepts designate a set of similar objects, events, or symbols with common characteristics. Examples of concepts are: *mammal, triangle, cloud*, etc.

Procedures designate a set of ordered actions aimed at a goal. Examples of procedures are: *to subtract two numbers, to draw a map, to write a summarization*, etc.

Attitudes correspond with a prevailing tendency to act in a certain way and arise in front of determined situations, objects, events or people. Examples of attitudes are: *to keep your clothes clean*, *to enjoy listening to classical music*, etc.

This classification is being used extensively by professionals in Education to describe the contents, the goals, and the methods of evaluation of courses and subjects [23]. For example, let us suppose that the objective of a class is to teach the students how to use a map. An analysis of what this objective involves can reveal de following elements: First, the students must learn the concept of *map*, the concept of *scale*, *symbols for road*, *forest*, etc. Next the students can learn the pro-

cedures *to locate my position*, *to define a path*, etc. The attitude to teach is that *a map as a useful tool in certain situations*. This attitude will help the students to use maps when the situation requires it, for example, when the planning of a holiday trip.

In summary, we can say that students learn concepts to create an ordered description of the world, procedures to know the possibilities of action related with these concepts, and learn a set of attitudes with which they face different circumstances in their life.

In previous work the authors have introduced the use of CPA for system design in Robotics [34,33]. In this paper, we propose CPA as a useful paradigm to create a human-centered view of the system architecture for two main reasons:

- CPA is used in education as a tool for analyzing and structuring the knowledge to be taught. The knowledge required to understand a tele-operated robot architecture can be analyzed and structured using the same tool.

For example, CPA helps to organize the complexity of the system description and to analyze what the relevant information is to be presented to the operator in every circumstance.

- We would like the operator to consider the robot an intelligent partner, a partner that can have considerable autonomy.

Human beings learn, since childhood, to understand others (parents, siblings, friends, teachers, etc.) as mental and intentional agents as themselves [32]. Moreover, it is known that humans, when interacting with obviously inanimate objects, have no problem to attribute mental states as if these objects had intentionality [8]. By using CPA this impression can be enhanced as the operator observes a robot which is managing concepts and procedures, similar to himself/herself, and the robot is showing attitudes through the actions it chooses, similar to humans.

In the following paragraphs we elaborate in more detail upon the three types of knowledge required to understand the system architecture. Note that learning is an iterative process; we start by learning concepts, followed by procedures and attitudes. This cycle is repeated, which allows us to extend and refine knowledge with new Concepts, Procedures and Attitudes.

2.1. Concepts

The idea of concept has been extensively studied in different disciplines and there is much material available in the literature, for example, a general survey is provided in [21], a classic introduction to the subject of concept learning is given in [5], and an example of using abstract concepts in Robotics can be found in [27].

In *CPA-seven-levels (CPA7L)* we use networks of concepts to describe the system and its environment. Using conceptual networks allows us to provide an ordered picture of the system and its environment in several abstraction levels. Fig. 1 shows the Operator World Model. The largest set (W0) represents the network of all the concepts known to the operator. When the operator starts to work with the robot a learning process is initiated. During this learning process the operator creates relations with existing concepts in his/her mind using his/her perception of relevant physical objects, e.g., the workstation and its peripherals, the components of the robot, such as, motors, wheels, camera, etc. Additionally, the operator can learn more abstract concepts from the explanation given by a teacher. These concepts include details about the mission, details about robot maintenance, etc. This network of concepts (W1) includes all concepts the operator needs to know for robot operation.

The Operator World Model is further expanded by the information provided by the robot sensors, such as, ultrasonic sensors, infrared and video cameras, which the operator receives at the workstation. Thanks to these sensors the operator can incorporate new concepts into his/her world model, e.g. concepts like *obstacle*, *free-way*, *crossing people*, etc. These concepts are needed to understand the remote physical environment in which the mission takes place. These concepts (W2) are implemented with data structures the operator can access from the workstation and that also can be accessed by the autonomous robot behavior. An example of a shared concept is the map of obstacles the robot perceives using ultrasonic sensors and that the operator can see on the map that is displayed on the screen. A subset of W2 is W3, which is the Local World Model. Concepts in the Local World Model correspond with the immediate physical and temporal environment of the robot.

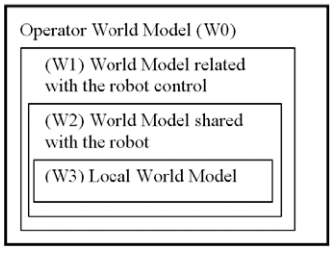

Fig. 1. Subsets of the operator world model.

It is the model of what is happening here and now. The concepts in W3 provoke the current behavior of the operator–robot team.

2.2. Procedures

The term procedure has a general accepted meaning. For our purposes the term procedure is closely related to the idea of *affordance* proposed by Gibson, where it refers to the action possibilities related with an object [14]. We use procedures to describe the possibilities of action related with the known concepts. Some of these possibilities of action are presented in the operator interface, some are used by the autonomous behavior of the robot, and some are shared.

Procedures are associated with the concepts in the World Model. The operator, who already had learned a first set of concepts, can learn procedures that enable him/her to use the robot to extend his/her own possibilities to act. The operator can learn not only the procedures related to the concepts in W1, e.g., the movement of the joystick or procedures related to the management of the workstation, but also procedures related to concepts shared with the robot in the remote environment, such as the procedures related to robot actuators. Examples of such procedures can be *to move the robot* , *to focus the camera*, *to measure the distance with ultrasonic sensors*, etc. The operator learns that for interacting with the environment, there are procedures that are operated by him/her but also procedures that can be controlled by the robot's autonomous behavior, i.e., procedures that are associated with concepts in W2.

It is worth noting that procedures are usually hierarchical; elemental procedures are related to the simplest actions and complex procedures are formed by a sequence of actions.

2.3. Attitudes

We have further developed the idea of Attitude as defined in the educational field, and provide it as a key element of the CPA paradigm. The operator learns Attitudes for knowing when and why to act as well as for understanding when and why the robot acts.

An important concept in the system architecture is the *degree of mission achievement* at every moment during the mission. All Attitudes in the system, that is, the operator Attitudes as well as the autonomous Attitudes of the robot, have the objective to increase this degree of mission achievement. In fact the top level attitude containing all others is called *to keep the degree of mission achievement as high as possible*.

Attitudes are built with two concepts, a concept *Origin* and a concept *Goal*, and one procedure, the *Plan*. *Origin* is a concept or network of concepts that correspond with a situation that could be detected in the Local World Model (W3). *Goal* is a concept that corresponds with a situation in which the *degree of mission achievement* is higher than in *Origin*. *Plan* is the set of procedures that can change the *Origin* situation into the *Goal* situation.

To illustrate an operator Attitude consider the following situation: the operator of an autonomous robot dedicated to bomb deactivation uses the feedback provided by a video image to guide the robot, searching for suspicious packages. The Attitude is *detect a bomb*, the *Origin* is the current situation with no bomb detected, the *Goal* is having a Local World Model containing the bomb, and the *Plan* is to use the camera and sensors to scan every room. In the same situation an example of an Attitude of the autonomous robot is *to keep the map updated*, to help the operator to navigate the building. The *Origin* is the old map. The *Plan* is to use ultrasonic sensors and odometry to update the local map. This local map is then matched to the old map, enabling the robot to locate itself in the environment. This updated map, the *Goal*, is presented to the operator.

Attitudes are triggered by the detection of *Origin* concepts in the Local World Model. An Attitude is a control loop that works iteratively to reach a *Goal*. Attitudes can have different levels of granularity. For example, an autonomous vacuum cleaner robot is designed to keep a set of rooms clean. *to keep the floor clean* could be the attitude corresponding to the general motivation for robot behaviour. Additionally we can consider sub-attitudes at the next level of detail, for example, *to maintain a map of dirty/clean areas*, *to avoid obstacles*, *to save the battery power*, *to optimize trajectories*, etc. all of which with subgoals to complete the top level Goal.

The system behavior is the result of the controlled fusion of the set of existing Attitudes.

3. A view of the architecture in seven levels

In this section, we use the CPA classification as the basis for creating a view of the system architecture, structured in seven hierarchical levels of functionality.

To some extent our proposal of this view is influenced by the OSI model for system interconnectivity [37]. Like the OSI model, CPA7L can be used to separate the functional parts of the system with clear interface boundaries between them. The OSI model provides an abstract description of layered communication, while CPA7L strives to do the same for tele-controlled robot architectures.

Fig. 2 shows the seven levels of the architecture (arrows represent relations of use). The higher level components add new functionality to the system using the resources provided by the components situated below. The seven levels allow us:

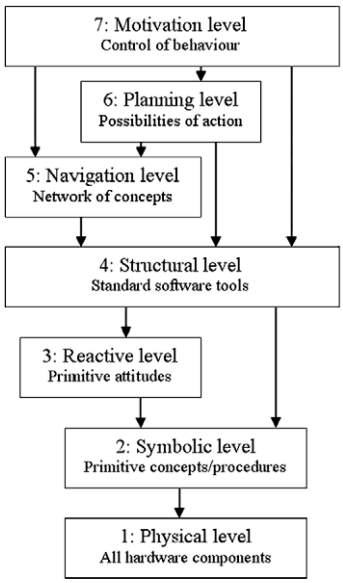

Fig. 2. A view of the architecture structured in seven levels including the relations of use.

(a) To organize the architecture of the computational system from the point of view of the system designer.
(b) To organize the information that the operator must know to perform specific tasks.
(c) To identify the seven parts of a complete HRI that could be implemented to facilitate the robot–operator communication.

In the following sections we will elaborate upon the components in the architecture, when viewed in the seven-level structure. The architectural view provided by *CPA7L* starts with the description of the system physical characteristics and continues, step by step, by including new levels of functionality until the whole system architecture is covered. In addition, a brief description of an implementation of the levels in a tele-operated robot system is included.

3.1. Physical level

This level contains a description of the physical elements of the robot and the operator workstation, which includes mechanical, electro-mechanical and electronic components of importance, such as sensors, actuators, as well as other parts of hardware of the processing system. The information at this level is enough for both the global understanding and (re)building of the physical parts.

Note that the concepts and procedures at this level belong to the Operator World Model (W1 in Fig. 1) and are inaccessible for the autonomous behavior of the robot.

For example: The Physical level description of the robot is in the form of text and diagrams, and contains information such as:

(a) Description of concepts: The robot is a tricycle with odometric sensors in the two motorized wheels. It has a video camera and two ultrasonic sensors mounted on servos. A WIFI board is included to make communication possible with the operator workstation. The main processor is a personal computer compatible board. Independent modules are included to control the ultrasonic sensors, the power supply and the motors. The operator workstation is a personal computer with WIFI connection capability, and a joystick.
(b) Description of procedures: The robot can navigate by providing different speeds to the two wheels. It is possible to measure the speed and position of the robot using odometers. The ultrasonic sensors can be moved independently and the distance between the nearest obstacle and the sensor can be obtained. The camera can be turned and focused, making it possible for the operator to see the robot environment.

As has been implemented in our prototype, the HRI at this level provides access to a detailed description of the robot physical characteristics. It is worth noting that at this level the operator can only obtain information that helps him/her to understand the system functioning and at this level the operator cannot interact with the system. He/she needs to select different tabs (see Fig. 4) in the interface to perform actions influencing the robot behavior.

3.2. Symbolic level

For processing information about the physical world it is necessary to establish a bridge between the physical world and the symbolic world [28]. The Symbolic level creates this bridge by providing the necessary mechanisms to access the Physical level from the symbolic context of the higher levels.

The Symbolic level includes a description of primitive concepts (memory registers) and primitive procedures (software primitive functions) which are used to access sensor measurements and to manage the actuators.

The operator can access this information in the format of a table where the physical concepts (for example the temperature measured by a sensor) are related with the names of variables (*Temperature*), indicating the position and size of the memory register where the processor can access this information. This table is an exhaustive detailed enumeration of the possibilities of primitive perceiving and acting. The concepts related to the values in this table (part of W2, see Fig. 1) can be manipulated by the operator by using the interface, or, by the autonomous behavior of the robot.

For example: The operator is explained that he/she has access to the motors through a Motor Control Module. This module gives access to two registers. The value written in the registers is proportional to the voltage applied to the motors, and therefore proportional to the rotation speed of the wheels (see Table 1). How much a wheel has turned is measured by the odometer. In our implementation one complete wheel rotation corresponds to an odometer value of 500. This odometer value is accessible through registers (Table 2). In Table 3 the set of basic procedures to command the motors is given. In our implementation the HRI provides access to the descriptions of these basic concepts and procedures. Furthermore these descriptive tables are currently accompanied by an interactive tool allowing the operator to directly read the input variables and write the output variables using the primitive functions. We introduced this functionality at this level to assist the robot maintenance expert in diagnostic tasks.

3.3. Reactive level

The Reactive level describes the mechanisms with which the tele-operated system reacts autonomously in real time, using the basic concepts and procedures provided by the Symbolic level. This component contains the system's basic attitudes. As in the previous levels, the elements at this level are usually highly optimized and cannot be easily modified (e.g. by learning) in a practical application.

For example, we can describe autonomous robots inspired by the Subsumption Architecture of Brooks, which are characterized by quick reactive responses and require no world model or planning [4], using only the first three levels in the CPA7L view of the architecture.

The Reactive level contains a set of basic attitudes that will be used by the attitudes at the 7th level. For example, the control of the gait of a legged robot could be built with a set of independent basic attitudes that control the different motors. These basic attitudes, in turn, are coordinated by a higher level attitude. A subset of attitudes is included aimed at protecting the robot from immediate threats (collision avoidance, excessive battery consumption).

For example: In our robot the PID controllers of the motorized wheels are used to implement low level attitudes. The PID controllers allow the robot to move smoothly, eliminating the otherwise jerky movements. These controllers are highly optimized and are implemented in the Motors Control Module. Another example of a basic attitude could be the control loop

Table 1
Basic concepts for using the motors.

Concept (physical signal)	Register description
Left motor voltage	1 byte. 0–127 back and 127–255 forward
Right motor voltage	1 byte. 0–127 back and 127–255 forward

Table 2
Basic concepts for getting feedback from the motors.

Concept (physical signal)	Register description
Left odometer	2 bytes. Fractions of turn
Right odometer	2 bytes. Fractions of turn

Table 3
Basic procedures for the motors.

Procedure	Concept
Set left motor voltage()	Left motor voltage
Read right odometer()	Right odometer

responsible for moving the servos on which the ultrasonic sensors are mounted. This control loop allows the robot to scan the environment and look for obstacles. This attitude is autonomously triggered while the robot is in movement.

The HRI in our implementation includes, at this level, a screen (tab) which provides a detailed description of the reactive attitudes. This helps the operator to be aware of the fact that when the operator is performing a mission, his/her actions on the tele-operated platform will be influenced by the reactive autonomous attitudes. In our implementation we provide the operator not only the possibility of real time monitoring of reactive attitudes but also direct access to reactive attitude parameterization. This option proves useful for maintenance purposes as well as during normal operation for understanding the robot behavior.

3.4. Structural level

As mentioned earlier, levels L1, L2, and L3 could be sufficient to describe a simple autonomous robot. Normally additional software infrastructure is needed for a more complex semiautonomous tele-operated system, such as, an operating system, a communication system, a database management system, etc. These elements are organized at the Structural level and provide the support needed for the higher levels.

A second role of the Structural level is to isolate higher level functionality from lower level functionality. As a result these higher level components can be reused on similar robotic platforms. Effectively the Structural level provides a generic interface that allows the higher levels to access the basic concepts, procedures and attitudes from the low level components.

For example: Our prototype uses two operating systems; on the robot Linux is installed and on the operator workstation Microsoft-Windows. The basic communication system between the operator and the robot provided by the operating systems includes a virtual terminal that can be accessed from the HRI at this level, providing access to the operating system commands.

3.5. Navigation level

The three next levels of the architecture description are dedicated to Navigation, Planning and Motivation, respectively. These levels describe how the computational system manages the three types of knowledge (Concepts, Procedures and Attitudes) used during the operator–robot collaboration. The basic Concepts, Procedures and Attitudes from lower levels are used as the starting point for building higher level representations of knowledge, as complex as the specific application requires.

The Navigation level contains a network of concepts corresponding to the world where the tele-operated robot executes the mission (W2 in Fig. 1). This World Model not only contains concepts related with the perceptions in the robot physical environment, but also concepts related with the perception of the physical elements that are part of the robot, as well as abstract concepts related with the mission. This network of concepts can be interpreted as a map representing where the operator and the robot navigate, with the goal of improving the *degree of achievement of the mission*.

For example, a World Model could include a map of a building, which would make it possible to find the best route to arrive to at a specific place. Note that the operator could collaborate by introducing restrictions while the autonomous behavior makes considerations about the expected power consumption, statistical data about the number of people in a specific corridor, etc.

The World Model maintained at this level is a resource to be used by the higher levels to obtain information about concepts. In systems with learning capability this level will need to provide functions to create, delete, and modify concepts.

The view of the architecture at this level contains the description of the whole conceptual network of the system, which makes it possible for the operator to navigate the structure of concepts and inspect the value of their attributes. Depending on the application the operator will be allowed to make more or less changes in the conceptual network. For example, an operator could be given access to the robot's map of a building and could be allowed to introduce meaningful textual labels related with the presented objects (names of rooms, chair, box, etc.). Such an interface could be directly used to expand the network of concepts shared with the robot.

For example: Our prototype uses two concepts to describe the robot physical state at every moment, namely, the robot speed and the battery level. The Local World Model includes a map with the obstacles surrounding the robot. This map is built with the help of the ultrasonic sensors that scan the front of the robot. The obtained distance values are transformed into an occupancy grid.

The HRI at this level is a graphical representation of the obstacles situated in front of the robot. Additionally the interface shows the current value of the speed and battery level of the robot.

3.6. Planning level

The view of the architecture at this level contains the procedures associated with the network of concepts available at the Navigation level. Using these procedures it is possible to create a possible plan of action that will lead the system from the current conceptual situation to a new situation with a higher degree of goal achievement.

Some of the procedures correspond with actions performed on the physical environment, e.g. *to cross a door*, *to follow a corridor* or *to push a box*. Other procedures perform transformations on abstract concepts, e.g. *to increase a counter* or *to add an obstacle to the map*.

The Planning level provides the Motivational level with a set of possibilities to act on the robot environment. Additionally, in an advanced tele-operation system, this level would include actions which make it possible to create, delete, and modify procedures and plans. Note that a system could possibly build plans autonomously by chaining simple procedures, for example: *to go back to the departure point*, *to connect the robot to the power supply*, *to calculate the optimal trajectory*, or, *to estimate the energy consumption in a mission*.

The HRI at this level can allow the operator to manipulate procedures in various ways, e.g. using a keyboard, a mouse, a joystick, or a more sophisticated haptic device [22]. Note that in these cases the operator is the agent that creates and executes action plans, using the resources provided by the architecture at this level.

For example: The HRI of our prototype gives the operator control over a procedure called *MoveRobot(vel, angle)*. The operator can command the robot trajectory using a joystick. The robot velocity and angle is proportional to the inclination and the angle of the joystick. Thus the interface provides the operator with the basic procedures on which he/she can execute his/her navigation plans in the physical environment.

3.7. Motivation level

The view of the system architecture at the Motivation level describes the expected system behavior in different circumstances. The view at this level includes the behavior of the operator, the possibilities of the autonomous robot behavior, and the description of how the collaboration between the two can be optimized. Note that these descriptions will be done by making reference to the concepts, procedures and basic attitudes introduced in the previous levels.

At this level the operator and autonomous robot behaviors appears as intelligent agents sharing resources provided by the lower levels to achieve the mission goals. In *CPA7L* the operator behavior is represented by a unique complex attitude that competes with the rest of attitudes for the control of the robot. It is worth noting that it could be useful to decompose the operator behavior into a set of attitudes representing his/her behavior when facing different circumstances.

An advanced tele-operated system, for example with a hierarchical structure of attitudes, could include possibilities to create, delete, and modify attitudes.

The interface at this level provides the operator with possibilities to act in collaboration with the robot. The operator perceives concepts (e.g. provided by the camera image) and uses procedures (e.g. on a joystick) and tries to attain one or more goals. At this level the architecture allows the operator to act as a collaborator influencing the system behavior in real time. The operator's own motivation acts as a complex attitude that collaborates with other available attitudes with the common goal *to increase the degree of mission achievement*.

Usually the control of the autonomous attitudes can be included in the HRI. For example, the operator interface could give the operator the possibility to enable or disable autonomous attitudes or the possibility to configure their parameters. An interesting option is to provide the operator with a system monitoring tool in the form of a system trace, showing the timing and activation of the set of attitudes during the mission development.

For example: Our prototype has five attitudes that represent the operator–robot motivation. The first two are reactive, namely, *Watch-battery-level* and *Collision-avoidance*. The third attitude is an autonomous one called *Image-quality-adaptation*. This attitude reduces the quality of the transmitted image when the bandwidth is limited in order to ensure the image is updated in real time [25]. The fourth attitude implements the capacity of building a map of the obstacles in the environment. The fifth attitude is provided by the operator himself/herself, acting directly on the robot speed and direction using a joystick to reach a goal that can be seen on the workstation screen.

3.8. Summary

Levels L1, L2, L3 and L4 correspond with the simplest components of the system. The operator workstation typically is a personal computer with operating system. Also the robot can be viewed as a personal computer with additional sensors, actuators, plus the basic software to manage this hardware (Fig. 3 shows a photo of the robot).

Levels L5, L6 and L7 make it possible to develop complex applications. Level L5 can contain important information about the world in which the mission takes place. For example, we can imagine a *surveillance robot* that keeps track of the details of every object in a museum using its memory. In Level L6 a set of possibilities to act is represented. For example we can imagine a bomb-disposal robot that *knows* procedures to work with many types of bombs in different circumstances. Level L7 contains, in a conveniently separated component, the additional information required to provide the system the capacity for a semiautonomous behavior, e.g., the motivation of the system to act. The high level attitudes in this component could be used to build the behavior of a simple semiautonomous robot or to build a complex system where the learning of concepts, procedures, and even new attitudes could take place.

Fig. 4 shows a hypothetical "exhaustive" interface containing all information available at all different levels of *CPA7L*. By selecting a tab the operator can choose between partial views of the system at the different levels. The *Application tab* allows the operator to access a screen devoted to a specific application.

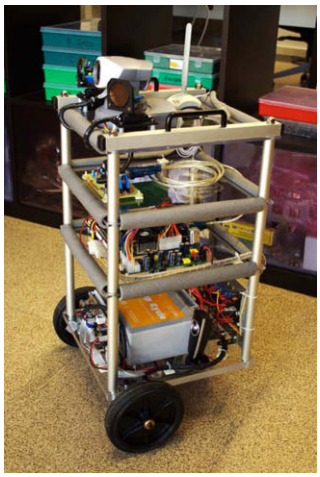

Fig. 3. Last version of our semiautonomous tele-operated robot.

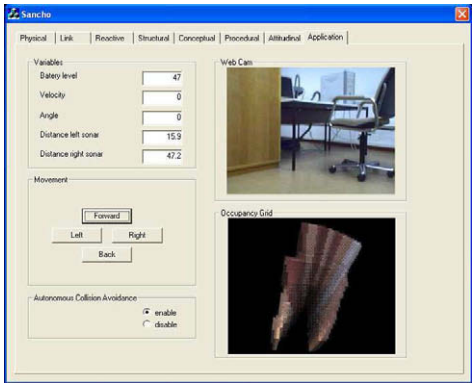

Fig. 4. A hypothetical interface which could be used to present all the different views of *CPA7L* to the operator.

4. Applications

The paradigm *CPA7L* presented earlier describes systems abstractly and can be applied in many different ways in the Robotics field. This section contains three different examples of application. The development of these examples is necessarily short and could be considered the starting point for future research. They are used here to give the reader an idea of the versatility of the *CPA7L* paradigm.

4.1. Use of CPA7L for teaching

The architectural description of a system should help the operator to establish a closer and more efficient collaboration with the robot. The following experiment shows how this can be accomplished using CPA7L.

We used the terminology of the *CPA7L* paradigm to explain briefly the system functioning to novice users. After this initial learning the users would operate the robot and were evaluated if they were able to operate the system properly. Additionally we would like to verify that the collaboration with the robot is perceived as natural by the students even when they know that they are working with an artificial object. The goal of this simple experiment was to test whether the operators learn to take advantage of the collaboration with the autonomous behaviors of the robot and check if better results were obtained.

The subjects in the experiment were eight students in the last year of their Master in Computing Science studies at the Polytechnic University of Madrid and had no prior knowledge of our project or our robot.

The student's task was to complete five laps in our laboratory (Fig. 5). The robot was controlled by the students from a separate room without direct visible contact. Students controlled the robot using a joystick and could view the robot camera image on screen. The time and the number of operator commands needed were measured for every lap. (The joystick software includes a low pass filter and sends a new command when the operator moves the joystick.)

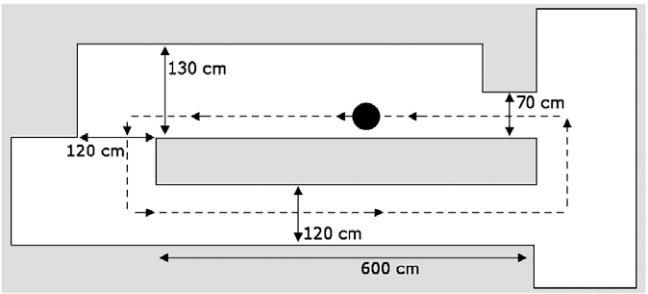

Fig. 5. Experimental lap in our laboratory.

The experiment was divided in two phases:

Phase 1: In the first phase the robot attitude for avoiding collisions was disabled. The students were taught the minimum required set of concepts and procedures to operate the robot. The disabled autonomous collision avoidance behavior was not mentioned. They were asked to complete five laps as fast as possible while avoiding collisions.
Phase 2: Once the first phase was finished the students were informed about the robot autonomy and about the fact that the robot can collaborate with them to complete the laps faster while avoiding obstacles.

This new information about the obstacle avoidance behavior was understood without any problem. Note that, with the explanation the students were given, they assumed that the robot was using its perception to get information and that the autonomous behavior of the robot was able to collaborate closely in guiding the robot path in real time, motivated by the common goal of completing the laps as fast as possible.

All the subjects finished both phases without causing collisions. Every subject obtained better lap times in the second phase. Fig. 6 shows that all the subjects reduced the number of commands given in the second phase.

4.2. Use of CPA7L in HRI analysis

In this second example we will see how the description of the architecture based on *CPA7L* can be useful for the HRI-analyst.

A well known model of human–computer interaction is GOMS, which has been the basis for a family of HCI-analysis techniques. In short we can say that the GOMS model is used to analyze the required knowledge of how to do a task in terms of Goals, Operators, Methods, and Selection rules [19].

The objective of these techniques is to predict the time a user needs to learn and use an interface as well as the level of internal consistency achieved by the interface. Recently, the GOMS model has been adapted to be applied to Robotics for the analysis of HRI [9].

– Goals are what the operator wants to accomplish by using the robot. Goals are often divided into subgoals. For example, to accomplish the goal of collecting some samples of mineral the operator might set subgoals: (1) to find the possible sources of mineral in a map of the territory, (2) to assess the nearest source in terms of distance and power consumption, (3) to command the robot to navigate towards the nearest resource, etc.

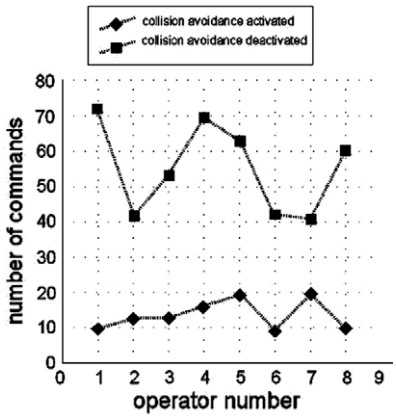

Fig. 6. The average number of commands used in robot control per lap by each operator.

- Operators are the actions the operator can do using the robot. With the original command–line interfaces, an operator was often obliged to type a command and its parameters using a keyboard. Nowadays, with graphic user interfaces, commands are just as likely to be menu selections, button presses, or direct-manipulation actions. Currently most of GOMS models define these operators at a concrete level, like button presses and menu selections.
- Methods are well-learned sequences of subgoals and operators that can accomplish a goal.
- Selection rules are the personal rules that users follow in deciding what method to use in a particular circumstance.

The *CPA7L* paradigm can be used side by side with GOMS techniques, i.e., once a *CPA7L* view of the architecture is available the work of a GOMS model designer will be considerably simplified. Working with semiautonomous robots the GOMS analyst must consider which part of the task will be performed by the operator, which part will be performed autonomously by the robot and which part will be done in collaboration.

- Goals: The view of the architecture at the Navigation level (L5) provides the analyst with a conceptual map where the operator goals can be clearly distinguished. The conceptual map required to define the operator's goals is a subset of the total conceptual network required to understand the complete architecture of the system and its mission environment. In this map we can distinguish the operators goals and the goals of the autonomous robot behavior. Moreover subgoals are the intermediate steps to go through when navigating this map of concepts.
- Operators and Methods are the procedures in *CPA7L*. The Operators and Methods available to the operator are a subset of the procedures described at the Planning level (L6).
- Selection rules are the operator's attitudes when faced with each specific circumstance in the development of his/her task.

The architectural view at the Motivational level includes a description of the set of existing attitudes. Using this description the GOMS analyst can analyze and determine which attitude will be activated in every circumstance. Moreover, when several attitudes are simultaneously activated, the analyst could study the degree with which every attitude participates in the accomplishment of a specific task.

4.3. CPA7L versus BDI

In the third example of application we propose using the *CPA7L* views of an architecture as a tool for describing functionality in terms of Beliefs, Desires, and Intentions (BDI). BDI is one of the best known models of practical reasoning agents. In short we can say that BDI was conceived with the intention to create an architecture of computational systems with the capacity of performing effective behavior in an uncertain dynamic environment [13].

- *Beliefs* represent knowledge of the world, i.e., the state of the world. In a dynamic environment this description will change continuously and in an uncertain environment this description is not necessarily true.
- *Desires* are the system goals. These goals represent some desired end state.
- *Intentions* are committed plans, i.e., procedures currently in execution. In a dynamically changing environment these procedures could be interrupted and substituted when needed.
- *Plans* are combinations of procedures, or parameterized procedures for use in future situations. In BDI these *Plans* are considered part of the *Beliefs*, i.e., they are part of the description of world.

The view of the architecture provided by *CPA7L* can be used as a basis for a description of the system in terms of the BDI model:

- *Beliefs* correspond straightforwardly with the idea of a map of interrelated concepts describing the mission environment, i.e., the World Model managed in L5. In BDI the elements in this map can change dynamically and the value of their attributes is defined in terms of probabilities or possibilities.
- *Desires* are the concepts in the network that represent goals.
- *Intentions* in BDI are related to attitudes in *CPA7L*. Attitudes are activated by the *Beliefs* about the current situation. Attitudes start the execution of *Intentions* (procedures) to achieve *Desires*.
- *Plans* can be a single procedure or a set of procedures grouped together for a purpose. In *CPA7L* the Planning Level contains a database of available procedures. Basic procedures are combined to create more elaborate procedures. This level contains the components to create the adequate *Plans* in every circumstance. In BDI as well as in *CPA7L* procedures are closely related to concepts. In *CPA7L* the related concepts are located in the World Model.

5. Conclusions

Current developments in semiautonomous tele-operated robots are the result of an interdisciplinary effort between different technical and scientific disciplines, namely, Software Engineering, Psychology, and Artificial Intelligence, among others.

The *CPA7L* paradigm results from the fusion of ideas coming from different disciplines. It provides a general, quite simple, human-centered view of the architecture of this type of systems.

CPA7L is defined at such a level of abstraction that it is possible to apply it in different ways in the Robotics field. It can be used to explain the architecture, to compare functionalities of different tele-operated robots, to design the HRI, or, to reuse an existing system by extending its functionalities.

CPA7L is a practical tool to face the challenge of developing the next generation of human-centered semiautonomous tele-controlled systems.

References

[1] R. Agarwala, A.W. Levinsona, The roboconsultant: telementoring and remote presence in the operating room using a novel mobile robotic interface, Urology 70 (5) (2007) 970–974.
[2] A. Birk, H. Kenn, Roboguard, a teleoperated mobile security robot, Control Engineering Practice 10 (11) (2002) 1259–1264.
[3] G. Booch, J. Rumbaugh, I. Jacobson, The Universal Modelling Language, Addison-Wesley Professional, 1999.
[4] R. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation 2 (1) (1986) 14–23.
[5] J. Bruner, J.J. Goodnow, G.A. Austin, A Study of Thinking, Science Editions, New York, 1967.
[6] P. Clements, F. Bachmann, L. Bass, Documenting Software Architectures: Views and Beyond, Addison-Wesley, Boston, 2003.
[7] C. Coll, Psicología y Curriculum, Paidos, 1994.
[8] D.C. Dennett, The Intentional Stance, Bradford Books/MIT Press, 1987.
[9] J.L. Drury, J. Scholtz, D. Kieras, Adapting GOMS to model human–robot interaction, in: Proceedings of the ACM/IEEE International Conference on Human–Robot Interaction, 2007, pp. 41–48.
[10] T. Fong, I. Nourbakhsh, Robots!: interaction challenges in human–robot space exploration, Interactions (ACM) 12 (2) (2005) 42–45.
[11] T. Fong, C. Thorpe, Vehicle teleoperation interfaces, Autonomous Robots 11 (1) (2001) 9–18.
[12] E. Garcia, P. Gonzalez de Santos, Hybrid deliberative/reactive control of a scanning system for landmine detection, Robotics and Autonomous Systems 55 (2007) 490–497.
[13] M. Georgeff, B. Pell, M. Pollack, M. Tambe, M. Wooldridge, The belief-desire-intention model of agency, in: Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98), vol. 1555, 1999, pp. 1–10.
[14] J.J. Gibson, The theory of affordances, in: R. Shaw, J. Bransford (Eds.), Perceiving, Acting, and Knowing, 1977.
[15] K. Goldberg, S. Gentner, C. Sutter, J. Wiegley, The Mercury Project: a feasibility study for internet robots, IEEE Robotics and Automation Magazine 7 (1) (1999) 35–40 (Special Issue on Internet Robotics).
[16] K. Han, S. Kim, Y. Kim, J. Kim, Internet control architecture for internet-based personal robot, Autonomous Robots 10 (2001) 135–147.
[17] N. Iwahashi, Language acquisition through a human–robot interface by combining speech, visual, and behavioral information, Information Sciences 156 (1) (2003) 109–121.
[18] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, Addison Wesley, Longman, 1999.
[19] B. John, D. Kieras, Using GOMS for user interface design and evaluation: which technique?, ACM Transactions on Computer–Human Interaction 3 (4) (1996) 287–319.
[20] N. Kubota, Computational intelligence for structured learning of a partner robot based on imitation, Information Sciences 171 (4) (2005) 403–429.
[21] S. Laurence, E. Margolis, Concepts and cognitive science, in: Concepts: Core Readings, 1999, pp. 3–81.
[22] J. Lee, J. Lim, C.W. Park, Development of ethernet based tele-operation systems using haptic devices, Information Sciences 172 (1–2) (2005) 263–280.
[23] A. Marchesi, E. Martín, Calidad de la enseñanza en tiempos de cambio, Alianza, 1999.
[24] M. Mataric, R.C. Arkin, Behavior-based Robotics, MIT Press, 1992.
[25] L. Mengual, J. Bobadilla, G. Trivino, A fuzzy multi-agent system for secure remote control of a mobile guard robot, Lecture Notes in Computer Science, vol. 3034, Springer-Verlag, 2004. pp. 44–53.
[26] R.R. Murphy, Human–robot interaction in rescue robotics, IEEE Transactions on Systems, Man and Cybernetics C 34 (2) (2004) 138–153.
[27] M. Persson, T. Duckett, A. Lilienthal, Virtual sensors for human concepts – building detection by an outdoor mobile robot, Robotics and Autonomous Systems 55 (5) (2007) 383–390.
[28] Z.W. Pylyshyn, Computation and Cognition: Toward a Foundation for Cognitive Science, MIT Press, Cambridge, MA, 1984.
[29] C.M. Reigeluth, M.D. Merrill, B.G. Wilson, R.T. Spiller, The elaboration theory of instruction: a model for sequencing and synthesizing instruction, in: Instructional Science, 1980, pp. 195–219.
[30] R. Simmons, J. Fernandez, R. Goodwin, S. Koenig, J. O'Sullivan, Learned from Xavier, IEEE Robotics and Automation Magazine 7 (2) (2000) 33–39.
[31] A. Lankenau, T. Rofer, Architecture and applications of the bremen autonomous wheelchair, Information Sciences 126 (1) (2000) 1–20.
[32] M. Tomasello, The Cultural Origins of Human Cognition, Harvard, New York, 1999.
[33] G. Trivino, J.C. Crespo, F. Fernández, The CPA paradigm for autonomous mobile robot knowledge representation, in: Proceedings of the International Conference on Signal Processing, Robotics and Automation, Cadiz, Spain, 2002.
[34] G. Trivino, A. Marchesi, Conceptos, procedimientos, actitudes y su aplicación en robótica, Infancia y aprendizaje 24 (2) (2001) 255–272.
[35] M. Wang, J. Liu, Interactive control for internet-based mobile robot teleoperation, Robotics and Autonomous Systems 52 (2–3) (2005) 160–179.
[36] Z. Zenn, H. Lee, Effective learning system techniques for human–robot interaction in service environment, Knowledge-Based Systems 20 (5) (2007) 439–456.
[37] H. Zimmermann, OSI reference model – the ISO model of architecture for open systems interconnection, IEEE Transactions on Communications 28 (4) (1980) 425–432.