

# Integrating human / robot interaction into robot control architectures for defense applications

Delphine Dufourd<sup>a</sup> and André Dalgalarondo<sup>b</sup>

<sup>a</sup>DGA / Service des Programmes d'Armement Terrestre  
10, Place Georges Clemenceau, BP 19, 92211 Saint-Cloud Cedex, France

<sup>b</sup>DGA / Centre d'Essais en Vol, base de Cazaux  
BP 416, 33260 La Teste, France

## ABSTRACT

In the near future, military robots are expected to take part in various kinds of missions in order to assist mounted as well as dismounted soldiers: reconnaissance, surveillance, supply delivery, mine clearance, retrieval of injured people, etc. However, operating such systems is still a stressful and demanding task, especially when the teleoperator is not sheltered in an armoured platform. Therefore, effective man / machine interactions and their integration into control architectures appear as a key capability in order to obtain efficient semi-autonomous systems. This article first explains human / robot interaction (HRI) needs and constraints in several operational situations. Then it describes existing collaboration paradigms between humans and robots and the corresponding control architectures which have been considered for defense robotics applications, including behavior-based teleoperation, cooperative control or sliding autonomy. In particular, it presents our work concerning the HARPIC control architecture and the related adjustable autonomy system. Finally, it proposes some perspectives concerning more advanced co-operation schemes between humans and robots and raises more general issues about insertion and a possible standardization of HRI within robot software architectures.

**Keywords:** Ground robotics, human robot interaction, control architectures, defense applications, reconnaissance robot.

## 1. INTRODUCTION

Given recent advances in robotics technologies, unmanned ground systems appear as a promising opportunity for defense applications. In France, teleoperated vehicles (e.g. AMX30 B2DT) will soon be used for mine breaching operations. In current advanced studies launched by the DGA (Délégation Générale pour l'Armement i.e. the French Defense Procurement Agency) such as PEA Mini-RoC (Programme d'Etudes Amont Mini-Robots de Choc) or PEA Démonstrateur BOA (Bulle Opérationnelle Aéroterrestre), robotics systems are considered for military operations in urban terrain as well as in open environments, in conjunction with mounted or dismounted soldiers. To fulfill the various missions envisioned for unmanned platforms, the needs for human / robot interaction (HRI) increase so as to benefit from the orthogonal capacities of both the man and the machine.

In section 2, we list several missions considered for military land robots and explain related issues concerning HRI. In section 3, we present an overview of existing paradigms concerning HRI and describe a few existing applications in the field of defense robotics. In section 4, we focus on an example concerning reconnaissance missions in urban warfare and present the work on the HARPIC architecture performed at the Centre d'Expertise Parisien of the DGA. Finally, in section 5, we raise more general questions about the introduction of HRI into control architectures and about a potential standardization of these interactions, before concluding in section 6.

## 2. OPERATIONAL CONTEXT

### 2.1. Missions for military robots

Unmanned ground systems can now be considered as key technologies for future military operations. Firstly, they remove humans from harms way by reducing their exposition on the battlefield. Moreover, they can provide various services and avoid humans dull, dirty or difficult tasks such as surveillance or heavy load carrying. More generally, in the future, they should be used as force multipliers and risk reducer in land operations.

---

Further author information:

D.D.: E-mail: delphine.dufourd@dga.defense.gouv.fr, phone: +33 (0)1 47 71 45 86

A.D.: E-mail: andre.dalgalarondo@dga.defense.gouv.fr, phone: +33 (0)5 57 15 47 62



**Figure 1.** A prospective view of future French network centric warfare.

As a result, the range of missions for unmanned ground systems in the defense and security field is widening. Among them, let us mention reconnaissance and scout missions, surveillance, target acquisition and illumination, supply delivery, mule applications, obstacle clearance, remote manipulation, demining, explosive ordnance disposal, retrieval of injured people, communication relays, etc. These missions include different time steps from planning and deployment, mission fulfilment with human supervision and possible re-planning to re-employment and maintenance, as well as training sessions. They take place on various terrains, ranging from structured urban environments to destructured (destroyed) places and ill-structured or unstructured open areas. Moreover, defense operations present a few specificities compared to civil applications. Unmanned vehicles will often have to deal with challenging ill-known, unpredictable, changing and hostile environments. In urban warfare for instance, robots will face many uncertainties and they will have to intermix with friends, foes and bystanders. As a partner or as an extent of a warfighter, a robot must also conform to different constraints and doctrines: he must neither increase the risk for the team – which stands for "surprise effect" cancelling and trap triggering for instance – nor impose an important workload to the human supervisor. Moreover, some decisions such as engaging a target cannot be delegated to a robot: it is crucial that man stay in the loop in such situations. Finally, robots will be inserted into large organizations (systems of systems designed for network-centric warfare – cf. Fig. 1) and will have to co-operate with other manned or unmanned entities, following a command hierarchy. Therefore, efficient and scalable man / machine collaboration appears as a necessity.

## 2.2. Why introduce HRI into military operations ?

On the one hand, it is desirable that robots should operate as autonomously as possible in order to reduce the workload of the operator and to be robust to communication failures with the control unit. For example, rescue operations at the World Trade Center performed using the robots developed for the DARPA (U.S. Defense Advanced Research Projects Agency) TMR (Tactical Mobile Robotics) program showed that it was difficult for a single operator to ensure both secure navigation for the robot and reliable people detection.<sup>1</sup>

On the other hand, technology is not mature enough to produce autonomous robots which could handle the various military situations on their own. Basic capacities that have been shown on many robots include obstacle detection and avoidance, waypoint and goal oriented navigation, detection of people, detection of threats, information retrieval (image, sound), localization and map building, exploration or coordination with other robots. Most of them can be implemented on a real robot with a good reliability in a reasonably difficult environment but it is not enough to fulfill military requirements. For instance, so far, no robot has been able to move fully autonomously in the most difficult and unstructured arenas of the NIST (US National Institute of Standards and Technologies) in the context of search and rescue competitions.<sup>2</sup> Moreover, some high-level doctrines and constraints (such as discretion) may be difficult to describe and formalize for autonomous systems.

Finally, humans and robots have complementary capabilities<sup>3</sup>: humans are usually superior in perception (especially in environment understanding), in knowledge management and in decision making while robots can be better than humans in quantitative low-level perception, in precise metric localization and in displacement in confined and cluttered space. Above all, robots can stand dull, dirty and dangerous jobs. Therefore, it seems

that today's best solutions should rely on a good collaboration between humans and robots e.g. when robots act as autonomously as possible but remain under human supervision.

### 2.3. What kind of HRI is needed for military operations ?

The modalities concerning human / robot interaction may vary depending on the situation, the mission or the robotic platform. For instance, during many demining operations, the teleoperation of the robot could be performed with a direct view of the unmanned vehicle. However, the teleoperation of a tracked vehicle like the French demonstrator SYRANO (Système Robotisé d'Acquisition et de Neutralisation d'Objectifs) is performed with an Operator Control Unit (OCU) set up in a shelter which could be beyond line of sight if the transmission with the robot is ensured. In urban environment, the dismounted soldier in charge of the robot may need to protect himself and to operate the robot under stressful conditions either in close proximity or at a distance varying from a few meters to hundreds of meters. Concerning the effectors and the functions controlled by the human / robot team, some missions could imply precise remote manipulation with a robotic arm or the use of specific payloads (pan / tilt observation sensors, diversion equipments such as smoke-producing devices...) while others would only include effectors for the navigation of the platform. Finally, some missions may be performed in a multi-robot and multi-entities context, using either homogeneous or heterogeneous vehicles, with issues ranging from authority sharing to multi-robot cooperation and insertion into network-centric warfare organizations. Therefore, defining general HRI systems for robots in defense applications may be quite challenging.

## 3. EXISTING HUMAN / ROBOT INTERACTION SYSTEMS

### 3.1. Overview of existing paradigms for HRI

Many paradigms have been developed for human robot interaction. They allow various levels of interaction with different levels of autonomy and dependance. To introduce our work concerning the HARPIC architecture as well as related work in the defense field, we attempt to briefly list below the main paradigms from the least autonomous (teleoperation) to the most advanced (mixed-initiative) where humans and robots can co-operate to determine their goals and strategies.

*Teleoperation* is the most basic and mature mode. In this mode the operator has full control of the robot and must assume total responsibility for mission safety and success. This mode is suitable in complicated or unexpected situations which no algorithm can deal with. In return this mode often means a heavy workload for the teleoperator who needs to focus his attention on the ongoing task. With *Supervisory control*<sup>4</sup> the operator (called the supervisor) orders a subordinate (the robot) to achieve a predefined plan. In this paradigm, the robot is merely a tool executing tasks under operator monitoring. This interaction mode can adapt to low level bandwidth and high level control but needs constant vigilance from the operator who must react in case of failures (to perform manual mission re-planning for example). This mode is only suitable for static environments where planning is really effective. *Behavior-based teleoperation*<sup>5</sup> replaces fine-grained control by robot behaviors which are locally autonomous. In comparison to supervisory control, this paradigm brings safety and provides the robot with the opportunity to react to its own environment perception (e.g. to avoid obstacles). Thus, it allows the operator to be more negligent. *Adjustable autonomy*<sup>6</sup> refers to the adjustment of the level of autonomy which has been initiated by the operator, by another system or by the system himself and while the system operates. The goal of this paradigm is generally to increase the negligence time allowed to the operator while maintaining the robot to an acceptable safety and effectiveness level. In *Traded control*<sup>7</sup> the operator controls the robot during one part of the task and the robot behaves autonomously during the other part of this task. This paradigm may lead to an equivalent of the kidnapped robot problem. While the robot is controlled by the operator, it can lose its situation awareness and face difficulties when coming back to autonomous control. In *Shared control*<sup>8</sup> part of the robot functions are autonomous and the remaining are controlled by the operator. It is important to notice that this mode requires constant attention from the operator. If the robot is only in charge of some safety mechanisms, this paradigm is equivalent to the *safeguarded teleoperation*. *Mixed-initiative*<sup>9</sup> is characterized by a continuous dialogue between the operator and the robot where both of them share decision and responsibility. *Collaborative control*<sup>10</sup> may be described as a restrictive implementation of mixed-initiative. In this approach, the human robot dialogue is mainly reduced to a predefined set of questions and answers.

All these paradigms may be considered as subsets of the human / robot teaming domain with important overlapping. Therefore, many robotic experimentations use a mix of these paradigms, including our work on HARPIC and related projects described in the next section.

### 3.2. Existing applications concerning defense applications

Most applications for military unmanned ground vehicles are based on existing control architectures and the HRI mechanisms often reflect this initial approach. Most of these architectures are hybrid, but some of them are more oriented towards the deliberative aspect while others focus mostly on their reactive components. As a result, HRI mechanisms are sometimes more oriented towards high level interaction at the planning phase (deliberative) while others tend to be more detailed at the reactive level.

For instance, the small UGV developed by the Swedish Royal Institute of Technology<sup>11</sup> for military operations in urban terrain is inspired from the SSS (Servo, Subsumption, Symbolic) hybrid architecture. The individual components correspond to reactive behaviors such as goto, avoid, follow me, mapping, explore, road-follow and inspect. During most tasks, only a single behavior is active but when multiple behaviors are involved, a simple superposition principle is used for command fusion (following the SSS principles). The activation or deactivation of a behavior seems to be achieved using the planning realized before the mission. The interface is state-based (modeled as a simple finite automata) so that the user can choose a task from a menu and make simple use of simple buttons to control a task. However, the authors do not describe precisely the planning module (the deliberative level) nor its relationship with the direct activation of behaviors (more reactive). Thus, it seems that the stress has been laid on the lower levels of the architecture (Servo, Subsumption).

The software architecture of the US Idaho National Engineering and Environmental Laboratory (INEEL), which has been tested by the US Space and Naval Systems Center in the scope of the DARPA TMR (Tactical Mobile Robotics) project,<sup>12</sup> was partially inspired by Brooks' subsumption architecture, which provides a method for structuring reactive systems from the bottom up using layered sets of rules. Since this approach can be highly robust in unknown or dynamic environments (because it does not depend on an explicit set of actions), it has been used to create a robust routine for obstacle avoidance. Within INEEL's software control architecture, obstacle avoidance is a bottom-up layer behavior, and although it underlies many different reactive and deliberative capabilities, it runs independently from all other behaviors. This independence aims at reducing interference between behaviors and lowering overall complexity. INEEL also incorporated deliberative behaviors which function at a level above the reactive behaviors. Once the reactive behaviors are "satisfied", the deliberative behaviors may take control, allowing the robot to exploit a world model in order to support behaviors such as *area search*, *patrol perimeter* and *follow route*. These capabilities can be used in several different control modes available from INEEL's OCU (and mostly dedicated to urban terrain). In *safe mode*, the robot only takes initiative to protect itself and the environment, but allows the user to otherwise drive the robot. In *shared mode*, the robot handles the low-level navigation, but accepts intermittent input, often at the robot's request, to help guide the robot in general directions. In *autonomous mode*, the robot decides how to carry out high-level tasks such as *follow that target* or *search this area* without any navigational input from the user. Therefore, this system can be considered as a sliding autonomy concept.

The US Army Demo III project is based on the reference architecture 4D-RCS designed by the NIST (based on the German 4D and on the NIST former RCS architectures) which allows a decomposition of the robot mission planning into numerous hierarchical levels.<sup>13</sup> This architecture can be considered as hybrid in the sense that it endows the robot with both reactive behaviors (in the lower levels) and advanced planning capabilities. Theoretically speaking, the operator can explicitly interact with the system at any hierarchical level. The connections to the OCU should enable a human operator to input commands, to override or modify system behavior, to perform various types of teleoperation, to switch control modes (e.g. automatic, teleoperation, single step, pause) and to observe the values of state variables, images, maps and entity attributes. This operator interface could also be used for programming, debugging and maintenance. However, in the scope of the Demo III program, only lower levels of the architecture have been fully implemented (servo, primitive and autonomous navigation subsystems) but they led to significant autonomous mobility capacities tested during extensive field exercises at multiple sites.<sup>14</sup> The OCUs feature touch screens, map-based display and context-sensitive pull-down menus and buttons. They also propose a complete set of planning tools as well as built-in mission rehearsal and simulation capabilities. In the future, the NIST and the Demo III program managers intend to implement more tactical behaviors involving several unmanned ground vehicles, thus addressing higher levels of the 4D-RCS hierarchy. This will enable the developers to test effectively the scalability and modularity of such a hierarchical architecture.

The German demonstrator PRIMUS-D,<sup>15</sup> dedicated to reconnaissance in open terrain, is also compatible with the 4D/RCS architecture. The authors provide various details about the data flows between subsystems, which gives indications about the way HRI components could interact and be linked with robot subsystems. Within PRIMUS OCU, all in- and outputs are organized in the segment "Man-Machine Interface" which enables the operator to perform mission planning, command and control of the vehicle. Inputs from the operator flow to a "coordination" segment which performs mainly plausibility checks, command management and command routing.

Commands or entire mission plans are transmitted to the robot vehicle by the segment “communication”, if “coordination” decides that the command could be executed by the robot. An additional software module called “payload” manages the RSTA payload. Concerning the outputs from the robot, “communications” forwards received data depending on the data type to coordination (robot state information) or to “signal and video management”. On the robot side, command systems are received by a “communication” segment and forwarded to “coordination” for plausibility checks. If the command can be executed by the robot, the command will be pipelined to the next segment: either “payload” if it concerns the RSTA module or “driving” and “perception” if it is related to the navigation of the platform. The PRIMUS robot can be operated according to five modes: autonomous driving with waypoint navigation, semi-autonomous driving (with high-level commands such as direction, range and velocity), remote control (assisted with obstacle detection and collision avoidance), road vehicle and autonomous vehicle following. It can thus be considered as an adjustable autonomy concept, including simple or behavior-based teleoperation and autonomous modes. However, it is unclear how modular the system is and how new capabilities could be added to the system.

In the scope of the DARPA MARS (Mobile Autonomous Robotic System) program, Fong, Thorpe and Baur have developed a very rich HRI scheme based on collaborative control where humans and robots work as partners and assist each other to achieve common goals. So far, this approach has been mainly applied to robot navigation: the robot asks human questions to obtain assistance with cognition and perception during task execution. An operator profile is also defined so as to perform autonomous dialog adaptation. In this application, collaborative control has been implemented as a distributed set of modules, connected by a message-based architecture.<sup>10</sup> The main module of this architecture seems to be the safeguarded teleoperation controller which supports varying degrees of cooperation between the operator and the robot. Other modules include an event logger, a query manager, a user adapter, a user interface and the physical mobile robot itself.

Finally, concerning multirobot applications, the DARPA TMR program also led to the demonstration of multirobot capabilities in urban environment based on the AuRA control architecture, in relation with the MissionLab project.<sup>16</sup> This application relies on schema-based behavioral control (where the operator is considered as one of the available behaviors for the robots<sup>17</sup>) and on a usability-tested mission specification system. The software architecture includes three major subsystems: 1) a framework for designing robot missions and a means for evaluating their overall usability; 2) an executive subsystem which represents the major focus for operator interaction, providing an interface to a simulation module, to the actual robot controllers, to premission specification facilities and to the physical operator ground station; 3) a runtime control subsystem located on each active robot, which provides an execution framework for enacting reactive behaviors, acquiring sensor data, reporting back to the executive subsystem to provide situation awareness to the team commander. A typical mission starts with a planning step through the MissionLab system. This mission is compiled through a series of languages that bind it to a particular robot (Pioneer ou Urbie). It is then tested in faster than real-time simulation and loaded to the real robot for execution. During execution, a console serves as monitor and control interface: it makes it possible to intervene globally on the mission execution (stop, pause, restart, step by step...), on the robot groups (using team teleautonomy, formation maintenance, bounding overwatch, by directing robots to particular regions of interest or by altering their societal personality), and on the individual robots (activating behaviors such as obstacle avoidance, waypoint following, moving towards goals, avoiding enemies, seeking hiding places, all cast into mission-specific assemblages, using low-level software or hardware drivers such as movement commands, range measurement commands, position feedback commands, system monitoring commands, initialization and termination). The AuRA architecture used in this work can be regarded as mostly reactive: this reactivity appears mainly through the robot behaviors while other modules such as the premission subsystem look more deliberative (but they seem to be activated beforehand).

To conclude about these various experiences, most of these systems implement several control modes for the robot, corresponding to different autonomy levels: it seems that a single HRI mode is not sufficient for the numerous tasks and contexts military robots need to deal with. Most of them are based on well-known architectures (some of these architectures were originally designed for autonomous systems rather than semi-autonomous ones), leading to various HRI mechanisms. However, it is still difficult to compare all these approaches since we lack precise feedback (except for the few systems which were submitted to – heterogeneous but – extensive performance evaluations, e.g. teleoperation / autonomy ratio for Demo III,<sup>18</sup> MissionLab’s usability tests<sup>19</sup> or urban search and rescue competitions<sup>20</sup>) and since they address different missions and different contexts (mono-robot vs multi-robot, urban terrain vs ill-structured environments, etc.). Moreover, it is hard to assess their scalability and modularity in terms of HRI: for instance, does the addition of a new HRI mode compell the developer to modify large parts of the architecture, do they allow easy extensions to multi-robot and multi-operator configurations ?

In the next section, we will describe our work on the HARPIC architecture so as to illustrate and give a detailed account of an adjustable autonomy development: this description will raise more general issues about scalability and modularity, thus leading to open questions concerning the insertion of HRI within robot software architectures.

#### 4. PRESENTATION OF OUR WORK ON HARPIC

In November 2003, the French defense procurement agency (Délégation Générale pour l'Armement) launched a prospective program called "PEA Mini-RoC" dedicated to small unmanned ground systems. This program focusses upon platform development, teleoperation and mission modules. Part of this program aims at developing autonomous functions for robot reconnaissance in urban terrain. In this context, the Centre d'Expertise Parisien (CEP) of the DGA has conducted studies to demonstrate the potentialities of advanced control strategies for small robotic platform during military operations in urban terrain. Our goal was not to produce operational systems but to investigate several solutions on experimental platforms in order to suggest requirements and to be able to build specifications for future systems (e.g. the demonstrator for adjustable autonomy resulting from PEA TAROT).

We believe that in short term robots will not be able to handle some uncertain situations and that the most challenging task is to build a software organization which provides a pertinent and adaptative balance between robot autonomy and human control. For a good teaming, it is desirable that robots and humans share their capacities along a two-way dialogue. This is the approximate definition of the mixed-initiative control mode but, given the maturity of today's robots and the potential danger for soldiers, we do not think that mixed-initiative could be the solution for now. Therefore adjustable autonomy, with a wide range of control modes – from basic teleoperation to even limited mixed-initiative – appears as the most pragmatic and efficient way.

Thus, based on previous work concerning our robot control architecture HARPIC, we have developed a man machine interface and software components that allow a human operator to control a robot at different levels of autonomy. In particular, this work aims at studying how a robot could be helpful in indoor reconnaissance and surveillance missions in hostile environment. In such missions, a soldier faces many threats and must protect himself while looking around and holding his weapon so that he cannot devote his attention to the teleoperation of the robot. Therefore, we have built a software that allows dynamic swapping between control modes (manual, safeguarded and behavior-based) while automatically performing map building and localization of the robot. It also includes surveillance functions like movement detection and is designed for multirobot extensions.

We first explain the design of our agent-based robot control architecture and discuss the various ways to control and interact with a robot. The main modules and functionalities implementing those ideas in our architecture are detailed. Some experiments on a Pioneer robot equipped with various sensors are also briefly presented, as well as promising directions for the development of robots and user interfaces for hostile environment.

##### 4.1. General description of HARPIC

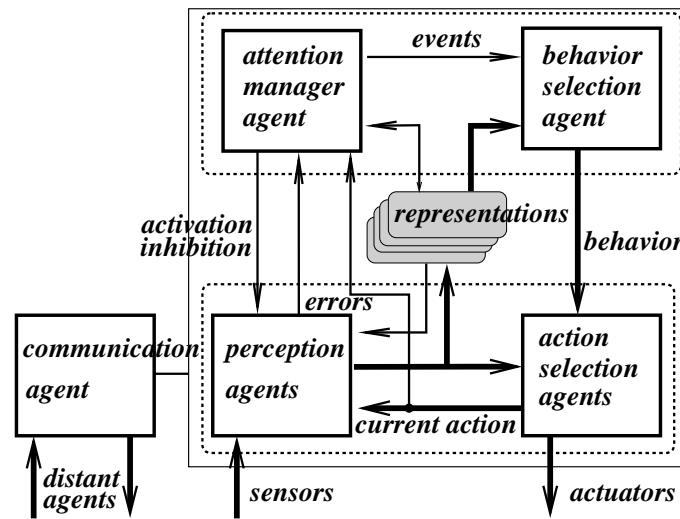
HARPIC is a hybrid architecture (cf. figure 2) which consists in four blocks organized around a fifth: *perception processes*, an *attention manager*, a *behavior selector* and *action processes*. The core of the architecture (the fifth block) relies on *representations*.

Sensors yield data to perception processes which create representations of the environment. Representations are instances of specialized perception models. For instance, for a visual wall-following behavior, the representation can be restricted to the coordinates of the edge detected in the image, which stands for the wall to follow. To every representation are attached the references to the process which created it: date of creation and various data related to the sensor (position, focus...). The representations are stored with a given memory depth.

The perception processes are activated or inhibited by the attention manager and receive information on the current behavior. This information is used to foresee and check the consistency of the representation. The attention manager has three main functions: it updates representations (on a periodical or on an exceptional basis), it supervises the environment (detection of new events) and the algorithms (prediction/feedback control), and it guarantees an efficient use of the computing resources. The action selection module chooses the robot's behavior depending on the predefined goal(s), the current action, the representations and their estimated reliability. Finally, the behaviors control the robot's actuators in closed loop with the associated perception processes.

The key ideas of this architecture are:

- *The use of sensorimotor behaviors binding perceptions and low-level actions both internally and externally:* the internal coupling allows to compare a prediction of the next perception (estimated from the previous perception and the current control) with the perception obtained after application of the control, in order to decide whether



**Figure 2.** Functional diagram of the HARPIC architecture.

the current behavior runs normally or should be changed; the external coupling is the classic control loop between perception and action.

- *Use of perception processes* with the aim of creating local situated representations of the environment. No global model of the environment is used; however less local and higher level representations can be built from the instantaneous local representations.
- *Quantitative assessment of every representation:* every algorithm is associated with evaluation metrics which assign to every constructed representation a numerical value expressing the confidence which can be given to it. We regard this assessment as important feature since any processing algorithm has a limited domain of validity and its internal parameters are best suited for some situations only. There is no perfect algorithm that always yields “good” results.
- *Use of an attention manager:* it supervises the execution of the perception processing algorithms independently from the current actions. It takes into account the processing time needed for each perception process, as well as the cost in terms of computational resources. It also looks for new events due to the dynamics of the environment, which may signify a new danger or opportunities leading to a change of behavior. It may also trigger processes in order to check whether the sensors operate nominally and it can receive error signals coming from current perception processes. It is also able to invalidate representations due to malfunctioning sensors or misused processes.
- *The behavior selection module* chooses the sensorimotor behaviors to be activated or inhibited depending on the predefined goal, the available representations and the events issued from the attention manager. This module is the highest level of the architecture. It should be noted that the quantitative assessment of the representations plays a key role in the decision process of the behavior selection.

## 4.2. HARPIC implementation

Fundamental capacities of our architecture encompass modularity, encapsulation, scalability and parallel execution. To fulfill these requirements, we decided to use a multi-agent formalism which fits naturally our need for encapsulation into independent, asynchronous and heterogeneous modules. The communication between agents is realized by messages. Object oriented languages are therefore absolutely suited for programming agents: we chose C++. We use POSIX Threads to obtain parallelism: each agent is represented by a thread in the overall process. For us, multi-agent techniques is an interesting formalism and although our architecture could be implemented without them it led to a very convenient and scalable framework.

All the agents have a common structure inherited from a basic agent and are then specialized. The basic agent can communicate by sending messages, has a mailbox where it can receive messages and runs its own process independently from other agents. Initially, all present agents have to register to a special agent called the administrator which records all information about agents (name, type, representation storage address...). All these data can be consulted by any agent. Then, when an agent is looking for another one for a specific job to

do, it can access to it and to its results. It is for example what is happening when an action agent has to use a representation coming from a perception agent.

Perception and action agents follow the global scheme. The action agent is activated by a specific request coming from the behavior selection agent. The selection agent orders him to work with a perception agent by sending its reference. The action agent sends in turn a request to the proper perception agent. Perception agents are passive, they only run upon request, perform a one shot execution and then wait for a new message. Furthermore, a perception agent can activate another agent and build a more specific representation using its complementary data. Many action and perception agents run at the same time but most are waiting for messages. Only one behavior (composed of a perception agent and an action agent) is active at the same time. Within a behavior, it is up to the action agent to analyze representations coming from the perception agent and to establish the correct control orders for the platform.

The attention agent supervises the perception agents. It has a look-up table where it can find the types of perception agents it has to activate depending on the current behavior. It is also in charge of checking the perception results and of declaring new events to the behavior selection agent when necessary. The advantage of this organization is detailed in a previous paper.<sup>21</sup>

The selection agent has to select and activate the behavior suited for the robot mission. This agent may be totally autonomous or constitute the process that runs the human computer interface. In this work, it is the second case and this agent is detailed in paragraph 4.4.

We use two specific agents to bind our architecture to hardware. The first one is an interface between the software architecture and the real robot. This agent awaits a message from an action agent before translating the instructions into comprehensible orders for the robot. Changing the real robot requires the use of a specific agent but no change in the overall architecture. The second agent acquires images from the grabber card at a regular rate and stores them in computer memory which can be consulted by perception agents.

Finally, we use a specific agent for IP communication with distant agents or other architectures. This agent has two running loops: an inner loop in which it can intercept messages from other agent belonging to the same architecture, and an external loop to get data or requests from distant architectures. This agent supervises the (dis)appearance of distant agents or architectures. It allows the splitting of an architecture on many computers or the communication between several architectures. For example, this agent is useful when agents are distributed between the robot onboard computer and the operator control unit.

### 4.3. Agents for SLAM and path planning

Map building is performed by a perception agent that takes laser sensor data and odometry data as input and outputs a representation which contains a map of the environment made of registered laser scans. The map building technique used combines Kalman filtering and scan matching based on histogram correlation.<sup>22</sup> This agent is executed whenever new laser data are available (e.g. 5 Hz), but it adds data to the map only when the robot has moved at least one meter since the last map update.

Localization is performed by a perception agent that takes odometry, laser data and the map representation as input and outputs a representation containing the current position of the robot. In its current implementation, this agent takes the position periodically estimated by the mapping algorithm and interpolates between these positions using odometry data to provide anytime position of the robot. This agent is executed upon request by any other agent that has to use the robot position (e.g. mapping, planning, human computer interface...).

Finally, path planning is carried out by an action agent that takes the map and position representations as inputs and outputs motor commands that drive the robot toward the current goal. This agent first converts the map into an occupancy grid and using a value iteration algorithm it computes a potential that gives for every cell of the grid the length of the shortest path from this cell to the goal. The robot movements are then derived using gradient descent on this potential from the current robot position. The path planning agent is used in the *Navigation* control mode of the operator control unit (described below).

### 4.4. HCI agent

In this implementation, our human computer interface is a graphical interface managed by the behavior selection agent of HARPIC. It is designed to use a small touch-screen of 320x240 pixels such as the one that equipped most personal digital assistant (PDA). With this interface, the user has to select a screen corresponding to one of the control mode he wants to activate or a screen showing the environment measures and representations built by the robot. These screens are described below.

The *Teleop* screen corresponds to a teleoperation mode where the operator controls the translational and the rotational speed of the robot (see figure 3 (left)) by defining on the screen the end of the speed vector. The



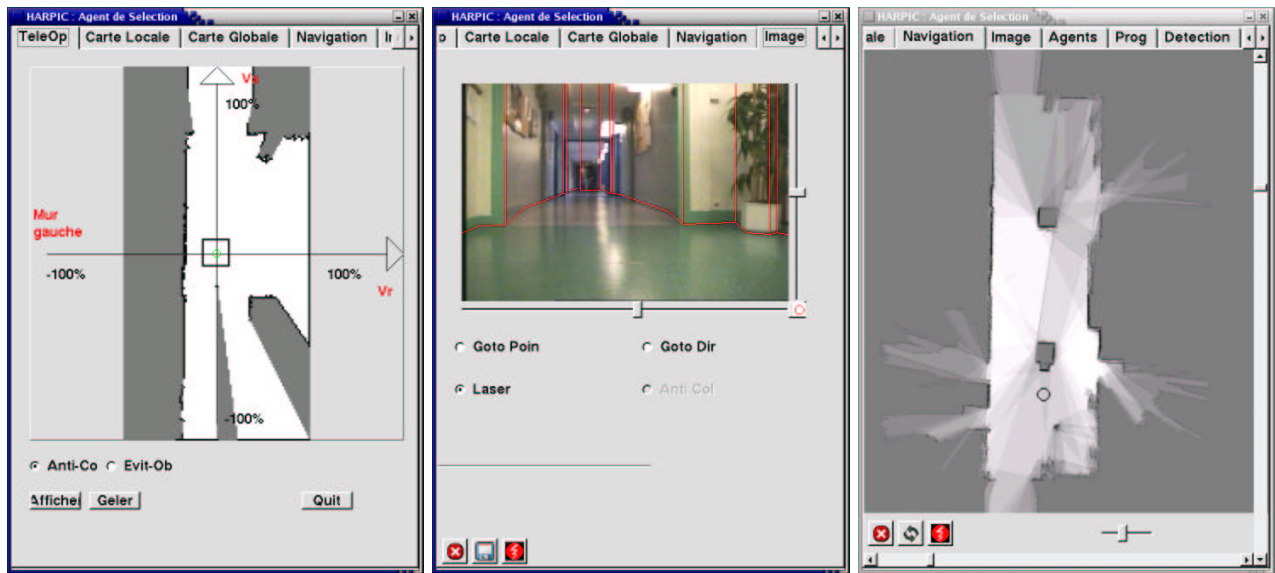


Figure 3. Interface for laser-based (left), image-based (center) teleoperation and goal navigation (right).

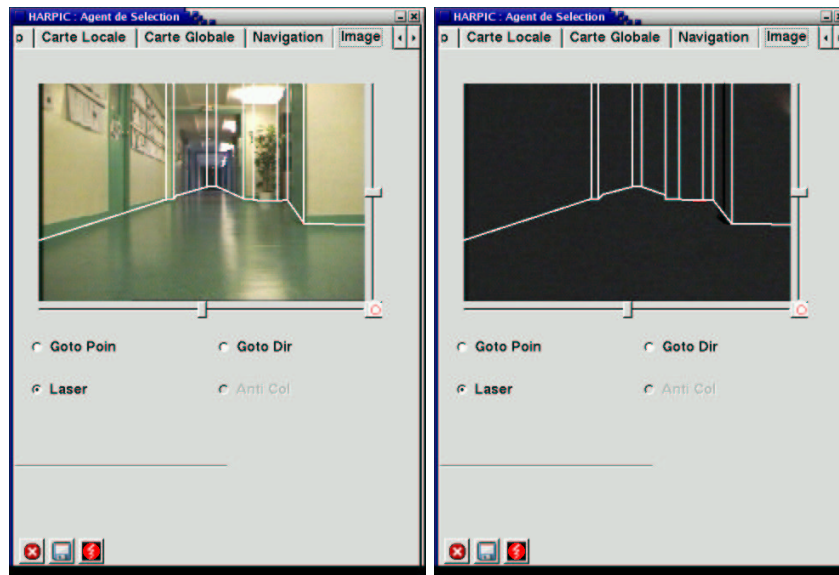


Figure 4. Image screen in normal (left) and in low-light condition (right) with overlaid polygonal view.

free space measured by the laserscanner can be superimposed on the screen and it appears in white color. The operator can also activate anti-collision and obstacle avoidance functions. Messages announcing obstacles are also displayed. This screen allows full teleoperation of the robot displacement (with or without seeing it thanks to the laser free space representation) as well as safeguarded teleoperation. As a result the operator has full control on the robot in order to trade with precise movement in cluttered space or with autonomous mobility failure. In return, he must keep defining the robot speed vector otherwise it stops and waits. A functionality related to the reflexive teleoperation<sup>12</sup> concept also enables the operator to activate behaviors depending on the events detected by robot. Indeed, clicking on “wall” or “corridor” messages appearing on the screen makes the system trigger “wall following” or “corridor following” behaviors, thus activating a new control mode.

The *Image* screen allows to control the robot by pointing at a location or defining a direction within the image acquired by the onboard robot camera. As the previous mode, this one enables full control or safeguarded teleoperation for the robot displacement. Two sliders operate the pan and tilt unit of the camera. When the *GoTo XY* function is enabled, the location selected by the operator in the image is translated into a robot displacement vector by projection onto the ground plane with respect to the camera calibration and orientation angles. The *Direction* function moves the robot when the operator defines the end of the speed vector on the

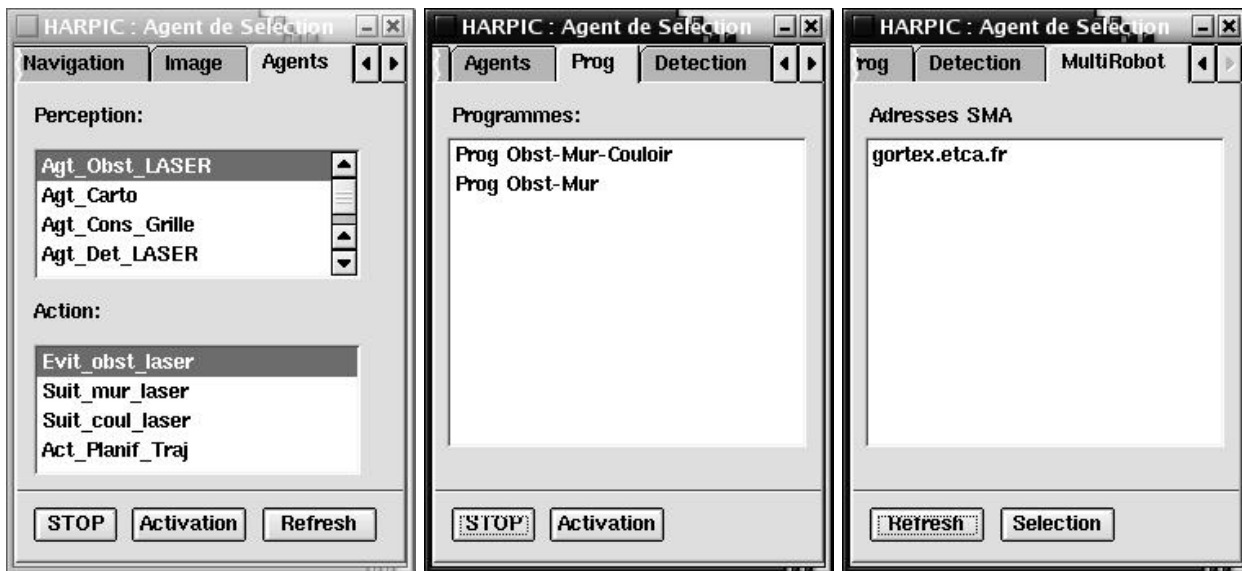


Figure 5. Interface for agent (left), program (center) and robot (right) selection.

screen. The selectable *Laser* function draws a polygonal view of the free space in front of the robot (in an augmented reality way) which is built from the projection of the laser range data in the image. This “Tron-like” representation allows to control the robot in the image whenever there is no sufficient light for the camera. Incidentally, this function provides a visual checking of the good correspondance between the laser data and the camera data. Figure 4 illustrates the effect of this function. If the *GoTo XY* or *Direction* functions are not enabled and if the robot is in any autonomous mode, this screen can be used by a operator to supervise the robot action by viewing the images of the onboard camera. However, it can stop the robot in case of emergency.

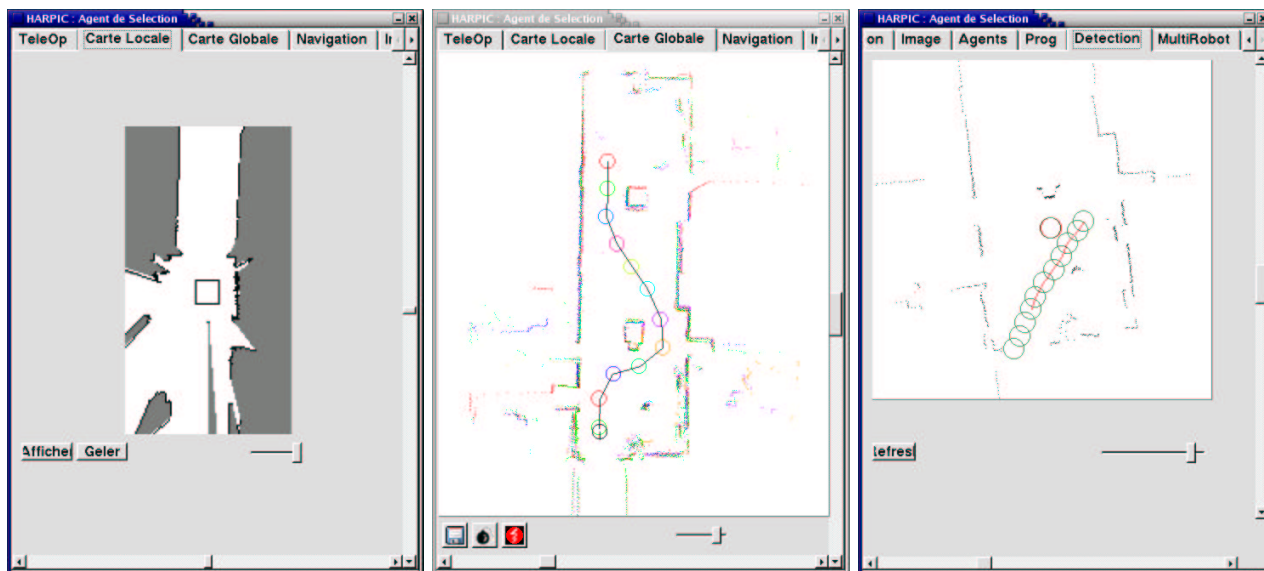
The *Navigation* screen shows a map of the area already explored by the robot. In this control mode, the operator has to point out at a goal location in the map to trigger an autonomous displacement of the robot up to this goal. The location can be changed dynamically (whenever the previous goal has not been reached). The planning process is immediate (a few seconds). When new areas are discovered the map is automatically updated. As shown in figure 3 (right), the map is an occupancy grid where bright areas correspond to location which have been observed as free more often than darkest area. Three sliders can translate or zoom the displayed map. This is an autonomous control mode where the operator can select a goal and forget the robot.

The *Agents* screen allows the composition and the activation of behaviors by selecting an appropriate pair of perception and action agents (see fig. 5 (left)). For example, it allows to execute behaviors like obstacle avoidance, wall following or corridor following with different perception or action agents (each one corresponding to a specific sensor or algorithm) for a same behavior. For example, a wall following behavior can result from a perception agent using the camera, from another one using the laserscanner and from various algorithms. This control mode corresponds to the behavior-based teleoperation paradigm. However this screen has been mainly designed for expert users and development purpose. It lacks simplicity but it will be easily reduced to a small number of buttons when the most effective behaviors set will be determined.

The *Prog* screen corresponds to predefined sequences of behaviors. For example, it enables the robot to alternate obstacle avoidance and wall or corridor following when respectively an obstacle, a wall or a corridor appears in the robot trajectory (see fig. 5 (center)). This example is a kind of sophisticated wander mode. More generally, this mode allows autoums tasks that could be described as a sequence of behaviors like exploration or surveillance where observation and navigation behaviors are combined. The list of these sequences can be easily augmented to adapt the robot to a particular mission. In this control mode, the robot is autonomous and the operator workload could be null.

A *MultiRobot* screen allows the operator to select the robot which will be controlled by his control unit. Indeed, our interface and software architecture is designed to adress more than one robot. In a platoon with many robots this capacity may give way to the sharing of each robot data or representation and to the exchange of robot control between soldiers.

The *Local Map* and *Global Map* screens show the results of the SLAM agents described in 4.3 (see fig. 6 (left and center)). The first one is a view of the free zone determined on each laser scan. The second one displays



**Figure 6.** Interface for local map (left), global map (center) and moving object detection (right).

the global map of the area explored by the robot and its trajectory. The circle on the trajectory indicates the location where the SLAM algorithm has added new data. The current position of the robot is also shown on the map. As on some others screens, sliders allow the operator to translate and to zoom the display. These screens may be used when the robot is in any autonomous mode to supervise its movements.

The *Detection* screen displays moving objects trajectory in the map built by the robot (see fig. 6 (right)). This screen stands for surveillance purpose. The algorithm used is based on movement detection on laser range data and Kalman filtering. This screen shows that this interface is not limited to displacement control of the robot but is able to be extended to many surveillance tasks.

The transition between any autonomous or teleoperation screens causes the ending of the current action or behavior. These transitions have been designed to appear natural to the operator. However, when one of these modes is activated, it is still possible to use the screens that display robot sensors data or representations without deactivating them. This feature is valid for the operator control unit but also for other control units. Thus, in a soldier team images and representations from a given robot can be viewed by a team member that is not responsible for the robot operation.

#### 4.5. Experimentation

The robot we used both in indoor and outdoor experiments is a Pioneer 2AT from ActivMedia equipped with sonar range sensors, color camera (with motorized lens and pan-tilt unit) and an IBEO laser range sensor. On board processing is done by a rugged Getac laptop running Linux and equipped with IEEE802.11 wireless link, frame grabber card and Arcnet card for connection to the laser (cf. figure 7). We also use another laptop with the same wireless link that plays the role of the operator control unit. The agents of the robot control architecture are distributed on both laptops. We did not use any specialized hardware or real-time operating system.

Several experiments have been conducted in the rooms and corridors of our building and have yielded good results. In such an environment with long linear walls or corridors, the autonomous displacement of the robot using the implemented behaviors is effective. However, in this particular space, a few limitations for the SLAM process have been identified. They mainly come from the laser measurement when the ground plane hypothesis is not valid and in the presence of glass surfaces.

The largest space we have experimented so far was the exhibition hall of the JNRR'03 conference.<sup>23</sup> Figure 8 shows the global map and the robot trajectory during this demonstration. It took place in the machine-tool hall of the Institut Français de Mécanique Avancée (IFMA) and in the presence of many visitors. As could be seen on figure 8, these moving objects introduced some sparse and isolated edges on the map but did not disturb the map building process. The robot travelled an area about  $60 \times 60$  m large with loops and abrupt heading changes. The robot displacement was mainly done in the safeguarded teleoperation mode because the building lacked main structures and directions for the robot to follow and because of the presence of people.



**Figure 7.** The experimental platform.

These experiments have revealed some missing functions in our interface (e.g. in mission map initialization, manual limitation of the space map, goto starting point behavior...) but no deep change requirement in the software architecture have been discovered.

#### 4.6. Future work

Our human computer interface runs on a PC laptop with Linux and Qt C++ toolkit for the graphic interface. It is currently being implemented on a PDA with Linux and Qtopia environment. Moreover, new functions and behaviors are being integrated onto the platform such as exploration, go-back-home and assistance in narrow space crossing like doors. More mission-oriented behaviors such as surveillance and people or vehicle recognition would also enhance our system and make it more directly suited to operational contexts. In the meantime, we keep improving existing behaviors to make them as robust as possible. Development of extended multirobot capacities, interaction and cooperation are also planned as a second step.

Concerning HRI, beyond considerations about ergonomics and usability of the interface, we consider working on semi-autonomous transition mechanisms between the various control modes, thus extending the simple reflexive teleoperation functionalities described above. These transitions could be triggered by changes in the environment complexity for instance: knowing the validity domain of the behaviors, it seems possible to activate more adapted behaviors, to suggest a switch towards another mode<sup>24</sup> or to request the help of the human operator. This would also lead to more sophisticated interaction modes such as cooperative control. On-line evaluation of the overall performance of behaviors or autonomy modes<sup>24</sup> also appears as a promising direction, all the more so as such evaluation mechanisms are already integrated within our architecture.<sup>25</sup>

Finally, it could be interesting to introduce more direct human/robot interactions for the various kinds of agents (beyond the interface agent). Perception agents might benefit from human cognition in order to confirm object identification for instance. Action agents might request human assistance when the robot cannot deal autonomously with challenging environments, while the attention agent could be interested by new human-detected events which would warn the robot about potential dangers or opportunities.

## 5. PERSPECTIVES AND OPEN ISSUES

The various HRI mechanisms existing in the literature, including our work on HARPIC, raise many general questions. For instance, generally speaking, how can we modify existing control architectures so as to introduce efficient HRI? What features would make some architectures more adapted to HRI than others? What kind of HRI is supported by existing architectures? Is it possible to conceive general standard architectures allowing any kind of HRI? What about scalability and modularity for HRI mechanisms? Which HRI modalities can be considered as most efficient within defense robotics applications? All these questions can still be considered as open issues. However, based on the examples described in the previous sections and on the recent advances in software technologies, we can provide a few clues concerning these topics.



**Figure 8.** Map and robot trajectory generated during a JNRR'03 demonstration in the IFMA hall. Each circle on the robot trajectory represents the diameter of the robot which is about 50 cm.

## 5.1. HRI within a robot control architecture

### 5.1.1. HRI modes

Humans and robots have different perception, decision and action abilities. Therefore, it is crucial that they should help each other in a complementary way. Depending on the autonomy control mode, the respective roles of the man and the robot may vary (see<sup>26</sup> for instance for a description of these roles according to the control mode). However, in existing interaction models, humans define the high-level strategies which are almost never transmitted to the robot: in the most advanced cases, the robot only knows task schedules or behaviors he must execute.

We have already described eight different HRI modes in section 3.1. Some variants such as safeguarded teleoperation or reflexive teleoperation could also be mentioned. Moreover, other approaches have been proposed to characterize autonomy, e.g. ALFUS<sup>27</sup> or ANS,<sup>28</sup> which could lead to other HRI mode definitions.

In the context of military applications, we have seen that adjustable autonomy can be considered as a promising mode. However, adjustable autonomy can lead to various mechanisms concerning control mode definitions and transition mechanisms between modes: these complicated issues are currently being addressed in the scope of PEA TAROT (Technologies d'Autonomie décisionnelle pour la Robotique Terrestre) for example.

### 5.1.2. Functions and functional levels concerned by HRI

Any function of the robot may be concerned by HRI, whether it be perception, decision or action. In any architecture, a human robot interface can theoretically replace any component receiving information and sending commands. However, in many cases, it is not meaningful to make such a replacement since some of these components can be dealt with (automated or computed) by machines very effectively. Indeed, the control of a mobile robot can globally operate at three different functional levels. At the lower level, it represents a direct control on effectors with sensory feedback and/or a direct view on the robot (teleoperation). At the next level,

the operator is in charge of the selection of tasks or behaviors. The upper level corresponds to the supervision where the operator takes part in the mission planning and monitors its execution. Depending on the design approach (respectively reactive, behavioral or deliberative), a control architecture can be modified in order to integrate a specific HRI operating either on the actuator commands, on the behaviors or on the mission planning/execution. This order might be difficult to bypass. For instance, on a very deliberative architecture including a planning module dedicated to the robot management, it would not be desirable that a HRI could enable an operator to intervene at the actuator or the behavior level (if the latter exists). Indeed, it might seriously disturb the planning execution (leading to the kidnapped robot syndrom).

Thus, within some architectures, interaction with other agents is only allowed at the highest levels. For instance, an extension of the three-tier architecture to the RETSINA MAS<sup>2</sup> only plans communications with the high-level reasoning layer. In specific contexts such as HRI in close proximity, the LAAS is currently developing sophisticated and dedicated interaction mechanisms (based on common human/robot goals) within the decisional layer of its architecture.<sup>29</sup> In DORO control architecture,<sup>30</sup> the user can interact with the higher module (the executive layer) during the planning phase (posting new goals, modifying planning parameters or the generated plan, etc.) and with the middle functional layer during the executive phase (e.g. deciding where the rover has to go or when some activities should stop). Nevertheless, the operator is not supposed to communicate with the lower physical layer. On the other hand, in simple reactive architectures like DAMN, command arbitration allows behaviors as diverse as human teleoperation and robot autonomous safety modules to co-exist.<sup>31</sup> Finally, in a hybrid architecture such as Harpic, where behavior chaining or planning are only considered as an advice, there is no major obstacle to the introduction (or the modification) of an HRI at any functional level. In general hierarchical architectures like NASREM,<sup>32</sup> 4D/RCS<sup>13</sup> or 3T,<sup>6</sup> HRI is also planned at every level so that autonomous behaviors can possibly override human commands or conversely be interrupted through human intervention.<sup>31</sup>

Moreover, we can notice that hybrid architectures seem well adapted to the insertion of HRI thanks to the concept of sensorimotor behaviors, which are usually quite meaningful for humans. These behaviors also look like the “elementary actions” of the soldiers, which also makes them good candidates for defense applications.

### 5.1.3. Human / robot interfaces

Human / robot interfaces are key components of semi-autonomous robots and they can be considered as part of their architecture. Roughly speaking, interfaces must provide the operator with information (or raw measurements) collected by the robot sensors and allow him to send orders. The semantic level of information and commands may vary but today, they are still inferior to those currently manipulated by humans. The interface can however display global and high-level information built by other human actors (such as the tactical situation). More details can be found in<sup>26</sup> for instance concerning classical basic and additional services for human / robot interfaces.

## 5.2. Constraints and limitations for the HRI design

### 5.2.1. Security constraints

In any given architecture, it seems possible (and sometimes necessary) to replace, double or duplicate a component (receiving information and sending commands) by a human/robot interface in the case when the human operator expertise or responsibility is not possible to avoid, for instance to ensure people and goods security. For instance, a robot must not endanger people in its vicinity.

### 5.2.2. Technical constraints about communication bandwidth and real time

Every functional level can be associated with a frequency concerning order exchange and information feedback between the human operator and the robot, especially within dynamic environments. For instance, to control an actuator level using a video feedback, the communication flow between the human and the robot must be important and very steady. On the other hand, a control at the mission planning level can accommodate sparse and even irregular communications with low bandwidth. The choice of the software and hardware communication technologies between the interface and the command receiver/information provider systems will depend on this constraint.

### 5.2.3. Constraints concerning ergonomics and human capacities

In the multirobot case or more generally when robots outnumber humans, it is not conceivable that robots should only be controlled at the actuator level since human operators will suffer from a heavy workload. Higher level commands (at the behavior or mission levels) are necessary. Moreover, it might be useful to reduce (filter) the information provided to the interfaces (e.g. within limited reality concepts) and to endow robots with autonomous behaviors so that they can carry on the mission on their own when they have not received any order.

The notion of user profile could be also used to adapt the content of an interface as well as its level of control so that it becomes better suited to a given human operator.<sup>10</sup> This process may refer to adaptation to the operator preferences, capacities or states (using pulse monitoring, temperature...). The interface should also adapt to the environment, e.g. by simplifying displays, showing only the most important information in emergency situations or selecting the information depending on the context.

### 5.2.4. Hardware constraints

At the hardware level, a human / robot interface is made both of an information output and order input device. The standardization constraint of an interface will mainly lie in the fact that one must avoid to use devices whose integration and ergonomics have not been specifically designed and adapted to a particular application such as a pilot joystick containing all degrees of freedom and buttons corresponding to the robot functions. In the near future, the most standard interface will probably be composed of a keyboard and a screen (tactile screen) or possibly a simple joystick. A specific interface can be simulated by a standard one (i.e. a tactile screen), despite a reduced ergonomics and probably a decreasing efficacy of the operator when sending orders.

### 5.2.5. HRI dependence from missions, platforms, payloads and interface devices

Concerning general HRI modes, independence seems quite challenging since context may induce very different HRI needs. For instance, dismounted soldiers will probably not be able to bear as much workload as mounted soldiers sheltered in an armoured control unit. They may also operate in close vicinity with the robot which will lead to specific collaboration schemes due to security reasons for instance (which might be similar to service robotics approaches) while remote operation of robot will imply other relationships with the robot, with awareness problems similar to those encountered in urban search and rescue (USAR) challenges.

Each mission and each environment can be associated to specific platform, payload and HRI device categories. In a dynamic and hostile environment, a dismounted soldier should use a simple, intuitive and small interface, which will be quite different from the one used by his unit commander in his command and control shelter. However, all these equipments can be built based on the same software technologies and on common communication interfaces.

## 5.3. New software technologies for HRI

Like usual systems in everyday life, military systems are evolving from very centralized computing towards distributed computing. Future warriors will be equipped with numerous sensors, will wear "smart" clothes and will team with various semi-autonomous systems. They will rely on technologies such as "ubiquitous, ambient and pervasive computing". Well-accepted and efficient computing needs to be context-sensitive, taking into account individual, collective as well as social dimensions.

Generally speaking (beyond robotic applications), multi-agent systems (MAS) enable coordination between distributed operations, creation of open systems with increasing complexity and increasing abstraction levels. This paradigm was developed around notions such as autonomy, interaction, organization and emergence to emphasize the relationship between the user and the system. Practically speaking, MAS combine several emerging technologies: distributed computing such as "grid-computing" (investigating how to use distributed and dynamically accessible computing resources in an efficient way), artificial intelligence, constrained programming, learning, data mining, planning, scheduling and web semantics. Thus, MAS make it possible to move from a traditional, centralized, closed and passive structure to a decentralized, open and pro-active conception. They are characterized by notions such as extensibility, portability, robustness, fault or failure tolerance and reliability.

Real-time issues as well as complex, constrained and embedded systems are still not fully addressed in the MAS community. However, agents can be created to address specific capabilities. Concerning real-time issues, numerous studies propose anytime algorithms providing results which are all the more satisfying as the algorithm execution time increases. These algorithms can be interrupted anytime, the quality of the result depending on the time allocated to their execution. Within HARPIC, this capability has been implemented using several agents dedicated to the same task, corresponding to algorithms with different costs. The selection of a specific agent thus depends on the available computing time and on its cost.<sup>25</sup>

As a result, it seems that the numerous capacities and technologies involved in MAS are well suited to new operational concepts such as network-centric warfare, where humans and manned systems are supposed to interact with semi-autonomous and robotic entities. The multi-agent paradigm also appears as an interesting framework for the development of robot control architectures designed for defense applications. Indeed, multi-agent systems enable to build bottom-up applications, without being constrained with evolving technical requirements. The conception of military systems (or systems of systems) whose life cycle may turn out to be long could thus benefit from the multi-agent paradigm. Indeed, agents can be considered as services which can be used by the system depending on its current needs.

Multi-threading and object-oriented languages can also facilitate the development of software architecture for robots, especially in a multi-agent framework. Beyond the similarity between the object and the agent concepts, the object-oriented language C++ has allowed us to develop common structures for agent hierarchies, inheriting from mechanisms such as communication by messages between the agents. For example, without resorting to multi-agent systems, the conceptors of the CLARAty architecture (considered as NASA future architecture) have also opted for an object-oriented approach in order to facilitate code reusability and extension.<sup>33</sup> Indeed, this approach favors a hierarchical and modular decomposition of the system at different levels of abstraction. For instance, a class can provide a locomotion function which will be more and more specialized as it comes closer to the effectors (wheels, tracks...). This approach could also favour a hierarchical decomposition of the HRI, depending on the precision of the information or orders exchanged between both entities.

As for most parts of the software architecture, the development of HRI mechanisms will probably rely on various tools and tool sets. These tools are likely to be unified within integration platforms including both architecture and environment and containing all the necessary functionalities to create interfaces and interaction modes, ranging from specification assistance to simulation and validation tools (see<sup>16</sup> for instance).

#### 5.4. HRI scalability

Scalability can be viewed as an extension capability avoiding:

- extensive studies concerning the design of the architecture when adding new functionalities;
- incompatibilities, disturbances or dead-lock between system components;
- reduced performances which could be due to coordination issues between the system components or to communication bottlenecks.

To tackle (and possibly solve) this problem, military organizations rely on specialized yet adaptive unities for every activity (based on human adaptation capabilities), communication and local negotiation or re-organization capabilities for every component, as well as a strong hierarchy wherein orders and reports flow rapidly.

Specialization, communication and hierarchy are quite easy to reproduce on software systems. Adaptation (including learning), negotiation and re-organization remain more open issues. Some of these capabilities have been developed for communication purposes for example, where QOS (quality of service) or transmission bandwidth are negotiated depending on the quality of the medium. They are also at the core of an emblematic advanced project by IBM concerning "Autonomic Computing", where a MAS-based system aims at delegating to software systems part of their own management, using self-healing and self-analysing agents. Moreover, agent hierarchization and prioritizing mechanisms have already been implemented on MAS, which appear as a promising paradigm in terms of adaptation, negotiation and re-organization. Indeed, such capabilities are part of the motivations for creating MAS.

Concerning the information or the representations which are to be manipulated within future large systems and systems of systems (containing both humans and robots), it seems difficult today to plan everything in advance. It should be possible to define the detail level for every entity. The information which could be delivered by entities of higher, similar or lower hierarchical ranks should be submitted to queries and answers, according to context-dependent rules or hierarchical decisions. One could thus envision future military network-centric systems as some kinds of web-services endowed with hierarchical rights and improved security.

#### 5.5. HRI standardization

Current design approaches concerning robot software architectures may not exploit the full possibilities for HRI (e.g. a given architecture may not adapt to any HRI mode): the emergence of reference and standardized architectures might help to reduce these specificities. Moreover, standards for architectures and for associated HRI would probably facilitate the extension of current systems, the specification and development of new architectures as well as the validation of their design.



However, HRI standardization goes beyond the simple fact of using standard technologies. It requires modular conception and component scalability as well. Thus, important issues remain regarding HRI design. Is it possible to build generic logical views of control architecture with associated HRI components, including the communication mechanisms between these components? Is there a limitation in this standardization taking into account the wide range of missions for robots or the various HRI modes?

Concerning the specific components that should appear in generic logical views of architectures including HRI, it seems that depending on the approach, new components may arise. For instance, in the context of service robotics, LAAS has defined interaction agents (representing interacting humans at the robot decisional level) as well as task delegates (monitoring task execution towards the goal and the level of engagement of interaction agents through specific observers).<sup>29</sup> In the case of Fong's approach about collaborative control,<sup>3</sup> specific segments have also been added such as a query manager or a user adapter. The challenge would thus consist in building views which would remain general enough to avoid specific solutions (related to specific contexts) and at the same time that would remain detailed enough so as to be useful during the specification, development and validation phases of new architectures.

At the software level, the standardization of human / robot interfaces including communication with surrounding systems is not very different from the standardization of any human / machine interface communicating with a wider application: this subject has been extensively studied in the field of distributed computing. For instance, the 3-tier architecture aims at dividing an application into three layers: a presentation layer, a processing layer and a data access layer. The presentation layer, i.e. the display and the dialog with the user can be based on HTML (using a web navigator) or on WML (using a GSM). Architectures such as SOAP (service oriented architectures), which have become popular through Web-services, make it possible to create services (software components with a strong inner coherence) with weak external coupling. These architectures rely on standards such as XML (Extensible Markup Language) for data description, WSDL (Web Services Description Language) for the description of service interfaces, UDDI (Universal Description Discovery and Integration) for service index management and SOAP (Simple Object Access Protocol) for service calling.

In military information systems, standards (16 or 22 links, MIDS...) have already been developed to exchange numerical messages, concerning tactical situations for instance. In MAS, interaction languages between agents have been specified within the multi-agent community, e.g. KQML or ACL (Agent Communication Language) from the FIPA (Foundation for Intelligent Physical Agents).

These technologies and the associated mechanisms could be used for software standardization purposes, for instance within networks where robots would interact with human operators and various other systems. It seems that these standards could also be considered as references for future HRI purposes in the defense field.

For example, in a very prospective approach, similarly to common approaches used in the web-services, could it be interesting to integrate a service index at the battlefield network level, allowing the negotiation of services between all entities? Will we see robots and soldiers naturally sharing information, helping each other to do their job and fulfill their mission?

## 6. CONCLUSION

In this article, we first listed some operational missions which could involve unmanned vehicles and explained various needs for HRI. In the next section, we mentioned a few robotic systems designed for military applications and described the HRI component within their software architectures. Then we presented our work concerning the development of an effective software enabling to control a reconnaissance robot: we have created a robot control architecture based on a multiagent paradigm which allows various levels of autonomy and interaction between an operator and the robot. According to the state of the art in robot autonomy in hostile environment we think that *adjustable autonomy* is one of the most pertinent choice for the next generation of military robots. Our work shows that many control modes with seamless transitions can be fairly integrated in a quite simple operator control unit. It has also confirmed the good behavior of our software architecture HARPIC and the great advantage of the flexible multiagent approach when adding many functions. Moreover, this kind of interface is a good mean to experiment, evaluate or demonstrate autonomous robot behaviors and HRI, which will hopefully allow us to gather more military requirements and to improve the specification of future systems. Finally, based on the lessons learned from the development of this software architecture and from other related work, we discussed more general issues concerning the adaptation of existing architectures to HRI and the possible standardization of HRI within reference architectures. We also proposed a few guidelines for the practical development of softwares for architectures which would be suited to HRI.

To conclude, military contexts offer a wide range of situations for robots and tackling these issues could lead to numerous innovations in the field of HRI. Even though some of these problems are currently addressed

in prospective programs launched by the DGA such as PEA Mini-RoC, PEA TAROT or PEA Demonstrateur BOA, there is still a major effort to be done in order to meet tomorrow needs.

## ACKNOWLEDGEMENTS

The work concerning HARPIC was entirely performed within the Geomatics-Imagery-Perception department of the Centre d'Expertise Parisien of the DGA. The authors would like to thank S. Moudere, E. Sellami, N. Sinegre, F. Souvestre, D. Lucas, R. Cornet, G. Calba, C. Couprie and R. Cuisinier for their contributions to the software development of this work. They are also very grateful to D. Filliat, M. Lambert, E. Moline and D. Luzeaux who shared their reflexions about autonomy and human / robot interaction and helped to improve the HARPIC system.

## REFERENCES

1. S. Pratt, T. Frost, A. M. Shein, C. Norman, M. Ciholas, G. More, and C. Smith, "Applications of TMR technology to urban search and rescue: lessons learned at the WTC disaster," in *SPIE AeroSense, Unmanned Ground Vehicle Technology IV*, (Orlando, USA), 2002.
2. I. R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-robot teaming for search and rescue," in *Pervasive Computing*, 2005.
3. T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, and human-robot interaction," in *Proceedings of the 10th International Symposium of Robotics Research*, (Lorne (Victoria, Australia)), November 2001.
4. P. Backes, K. Tso, and G. Tharp, "The web interface for telescience," in *IEEE International Conference on Robotics and Automation, ICRA '97*, (Albuquerque, NM), 1997.
5. M. Stein, "Behavior-based control for time-delayed teleoperation," Tech. Rep. 378, GRASP Laboratory, University of Pennsylvania, 1994.
6. G. Dorais, R. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost, "Adjustable autonomy for human-centered autonomous systems on Mars," in *6<sup>th</sup> International Joint Conference on Artificial Intelligence (IJCAI), Workshop on Adjustable Autonomy System*, 1999.
7. D. Kortenkamp, R. Bonasso, D. Ryan, and D. Schreckenghost, "Traded control with autonomous robot as mixed initiative interaction," in *14<sup>th</sup> National Conference on Artificial Intelligence*, (Rhode Island, USA), July 1997.
8. T. Röfer and A. Lankenau, "Ensuring safe obstacle avoidance in a shared-control system," in *Seventh International Conference on Emergent Technologies and Factory Automation, EFTA '99*, (Barcelona, Spain), 1999.
9. G. Ferguson, J. Allen, and B. Miller, "TRAINS-95: Towards a mixed-initiative planning assistant," in *Third International Conference on AI Planning Systems, AIPS-96*, 1996.
10. T. Fong, C. Thorpe, and C. Baur, "Collaborative control: a robot-centered model for vehicle teleoperation," in *AAAI Spring Symposium on Agents with Adjustable Autonomy, Technical report SS-99-06*, (Memlo Park), 1999.
11. H. I. Christensen, J. Folkesson, A. Hedström, and C. Lundberg, "UGV technology for urban navigation," in *SPIE Defense and Security Conference, Unmanned Ground Vehicle Technology VI*, (Orlando, USA), 2004.
12. E. B. Pacis, H. R. Everett, N. Farrington, and D. Bruemmer, "Enhancing functionality and autonomy in man-portable robots," in *SPIE Defence and Security Conference, Unmanned Ground Vehicle Technology VI*, (Orlando, USA), 2004.
13. J. Albus, "4D/RCS: a reference model architecture for intelligent unmanned ground vehicles," in *SPIE AeroSense Conference, Unmanned Ground Vehicle Technology IV*, (Orlando, USA), 2002.
14. J. A. Bornstein, "Army ground robotics research program," in *SPIE AeroSense Conference, Unmanned Ground Vehicle Technology IV*, (Orlando, USA), 2002.
15. G. Schaub, A. Pfaendner, and C. Schaefer, "PRIMUS: autonomous navigation in open terrain with a tracked vehicle," in *SPIE Defense and Security Conference, Unmanned Ground Vehicle Technology VI*, (Orlando, USA), 2004.
16. R. C. Arkin, T. R. Collins, and Y. Endo, "Tactical Mobile Robot mission specification and execution," in *Mobile Robots XIV*, pp. 150–163, (Boston, MA), September 1999.
17. K. S. Ali and R. C. Arkin, "Multiagent teleautonomous behavioral control," in *Machine Intelligence and Robotic Control (MIROC)*, **1(1)**, pp. 3–17, 2000.
18. B. A. Bodt and R. S. Camden, "Technology readiness level 6 and autonomous mobility," in *SPIE AeroSense Conference, Unmanned Ground Vehicle Technology VI*, (Orlando, USA), April 2004.

19. Y. Endo, D. C. MacKenzie, and R. C. Arkin, "Usability evaluation of high-level user assistance for robot mission specification," in *Special Issue on Human-Robot Interaction of the SMC Transactions: Part C*, 2004.
20. J. Scholtz, B. Antonishek, and J. Young, "Evaluation of human-robot interaction in the NIST reference search and rescue test arenas," in *Performance Metrics for Intelligent Systems (PerMIS'04)*, 2004.
21. D. Luzeaux and A. Dalgalarondo, *Hybrid architecture based on representations, perception and intelligent control*, ch. Studies in Fuziness and Soft Computing: Recent Advances in Intelligent Paradigms and Applications. ISBN-3-7908-1538-1, Physica Verlag, Heidelberg, 2003.
22. T. Röfer, "Using histogram correlation to create consistent laser scan maps," in *IEEE International Conference on Robotics Systems (IROS-2002)*, (EPFL, Lausanne, Switzerland), 2002.
23. "Quatrièmes Journées Nationales de Recherche en Robotique (JNRR'03)," (Clermont-Ferrand, France, <http://lasmea.univ-bpclermont.fr/jnrr03/>), 8-10 October 2003.
24. M. Baker and H. A. Yanco, "Autonomy mode suggestions for improving human-robot interaction," in *IEEE Conference on Systems, Man and Cybernetics*, October 2004.
25. A. Dalgalarondo, *Intégration de la fonction perception dans une architecture de contrôle de robot mobile autonome*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, janvier 2001.
26. A. Dalgalarondo, "De l'autonomie des systèmes robotisés," in *Technical report, ref. CTA/02 350 108/RIEX/807*, 2003.
27. H.-M. Huang, K. Pavsek, J. Albus, and E. Messina, "Autonomy levels for unmanned systems (ALFUS) framework: An update," in *SPIE Defence and Security Conference, Unmanned Ground Vehicle Technology VII*, (Orlando, USA), 2005.
28. G. M. Kamsickas and J. N. Ward, "Developing UGVs for the FCS program," in *SPIE AeroSense, Unmanned Ground Vehicle Technology V*, (Orlando, USA), 2003.
29. R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Joint Soc-EUSAI conference*, October 2005.
30. A. Finzi and A. Orlandini, "A mixed-initiative approach to human-robot interaction in rescue scenarios," in *International Conference on Automated Planning and Scheduling (ICAPS), Printed Notes of Workshop on Mixed-Initiative Planning and Scheduling*, pp. 36–43, 2005.
31. T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," in *IEEE Transactions on Industrial Electronics*, **50(4)**, August 2003.
32. J. Albus, R. Lumia, J. Fiala, and A. Wavering, "NASREM: the NASA/NBS standard reference model for telerobot control system architecture," in *Technical report, Robot System Division, National Institute of Standards and Technology*, 1987.
33. F. Ingrand, "Architectures logicielles pour la robotique autonome," in *Quatrièmes Journées Nationales de Recherche en Robotique (JNRR'03)*, (Clermont-Ferrand, France), October 2003.