Socially Interactive Robots for Real Life Use

**F. Michaud, C. Côté, D. Létourneau, Y. Brosseau, J.-M. Valin, É. Beaudry, C. Raïevsky, A. Ponchon,
P. Moisan, P. Lepage, Y. Morin, F. Gagnon, P. Giguère, A. Duquette, J.-F. Laplante, M.-A. Roux,
M.-A. Legault, T. Salter, S. Caron, P. Frenette, M. Lauria, F. Kabanza, P. Masson**
Université de Sherbrooke
Sherbrooke (Québec) CANADA J1K 2R1
{laborius-challenge}@listes.USherbrooke.ca

Abstract

Designing interactive mobile robots is a multidisciplinary endeavor that profits from having people interact with robots in different contexts, observing the effects and impacts. To do so, two main issues must be addressed: integrating perceptual and decision-making capabilities in order to interact in meaningful and efficient ways with people, and the ability to move in human settings. This paper describes four robotic platforms demonstrated at the AAAI 2005 Robot Competition, each addressing these issues in their own ways.

Introduction

We believe that socially interactive mobile robots must be designed by adopting a holistic approach, i.e., by addressing all design dimensions, from mechatronics to perceptual, decision-making and the intended application. In this paper, we briefly describe four mobile robotic platforms designed following this perspective, with an emphasis on Spartacus, our robot entry to the AAAI Mobile Robot events. References for more detailed explanations on the research work conducted with these robots are provided.

Roball, a Spherical Robotic Toy Used in Child Development Studies

A robot that can move in home environments, filled with all kinds of obstacles, requires particular locomotion capabilities. A mobile robotic toy for toddlers would have to move around other toys and objects, and be able to sustain rough interplay situations. Encapsulating the robot inside a sphere and using this sphere to make the robot move around in the environment is one solution. The encapsulating shell of the robot helps protect its fragile electronics. The robot, being spherical, can navigate smoothly through obstacles, and create simple and appealing interactions with toddlers (Michaud & Caron 2000). For instance, the robot can be programmed to wander around in the environment, moving away from obstacles sensed when making contact with them and generate different vocal reactions when being spun, shaken or pushed.

Figure 1: Roball-2.

Roball's second prototype, shown left in Figure 1, was specifically developed to be a toy. It was used to study interactions between a robot and toddlers using quantitative and qualitative evaluation techniques (Michaud *et al.* 2005a). Observations confirm that Roball's physical structure and locomotion dynamics generate interest and various interplay situations influenced by environmental settings and the child's personality. Roball is currently being used to see how child interaction can be perceived directly from onboard navigation sensors. Looking at sensory patterns over small periods of time, it is possible to distinguish what type of interaction the robot is receiving (e.g., being carried, being pushed, receiving no interaction, high levels of interaction, touching an obstacle, etc.), and adapt to this (Salter *et al.* 2005). Adaptation will first be kept simple, as it will be a proof-of-concept trial. It is likely to include the following: (1) Robot speeding up to high interaction levels and slowing down for low interaction, (2) Vocal messages responding to being picked up, spun and receiving no interaction.

Figure 2: Tito.

Tito and Children with Autism

Autonomy is not always a requirement for interactive robots. To study the use of mobile robots as pedagogical tools for children diagnosed with autism, we designed Tito. Shown right in Figure 2, Tito is a teleoperated wheeled robot of approximately 60 cm tall. It is colored (red, yellow, and blue) and its clothes are washable and made of soft material. It uses wheels to move, but its structure shows two feet and two legs to emulate a humanoid shape. It has two arms that can move up or down rapidly, a head that can rotate (to indicate 'no') and rise up (to express surprise), a mouth (for smiling), two eyes, a nose, hair (made from fiber optic cable to illuminate). Also, since we were interested in measuring eye gaze toward Tito, a small wireless microphone-camera was installed in one eye of the robot. Different parts of Tito's body can be illuminated and it is able to sense if it is being shaken or if it has flipped over. Tito generates vocal requests through pre-recorded messages. A wireless remote control (using a video game controller) was designed for teleoperation, and an on-board microcontroller enables preprogramed sequences of behaviors (motion and vocal messages). Examples of pre-programed behaviors are: moving the left arm while saying goodbye, expressing happiness by moving its arms, singing and rotating on the spot, shaking its head to indicate no, etc. Tito records and stores internally the timing between the interactions of the child (from sensory data and according to the experimental scenarios). Tito also emits a sound when it starts the execution of an experimental scenario, allowing synchronization of video data recorded with an external camera. Tito's activation button is placed underneath its base so that the child is not tempted to play with it. Tito was built in less than 6 months using modular distributed subsystems we had previously designed or used on other robots (e.g., energy monitoring, locomotion control, remote operation, sensing) (Michaud *et al.* 2005b).

Trials were conducted by a psycho-educator with Tito and four autistic children of approximately five years of age. Each child played individually with the robot in 10 minutes sessions, conducted three times per week over an eight week period. Overall, preliminary results (Duquette 2005) show that imitation of body movements and familiar actions occur more frequently with the experimenter than with the robot. However, attention sharing and imitation of facial expressions occur more frequently with the robot. Another interesting observation is that children figured out that Tito was teleoperated, which generated enthusiastic interactions between the child and the experimenter. This concurs with Robins et al. (Robins *et al.* 2004) observation that a robotic device can be used as a mediator for autistic children. In future work, our intent is to use Tito to identify useful adaptive capabilities that would allow Tito to autonomously interact with children.

AZIMUT, the Omnidirectional Modular Platform

AZIMUT, on the other hand, addresses the challenge of making multiple mechanisms available for locomotion on the same robotic platform. It is designed to operate in home-like environments, i.e., moving in tight areas, over different types of surfaces, dealing with various obstacles such as staircases (even circular ones), and work outdoors in all-terrain conditions. AZIMUT is symmetrical and has four independent articulations that can be wheels, legs or tracks, or a combination of these (Michaud *et al.* 2005b). By changing the direction of its articulations, AZIMUT is also capable of moving sideways without changing its orientation, making it omnidirectional (as shown in Figure 3). All these capabilities provide the robot with the ability to move in tight areas. Modularity is preserved at the structural level by putting the electrical and embedded systems inside the body of the robot, and by placing the actuators on the locomotion parts (i.e., Ferrite ServoDisc motors in the leg-track-wheel articulations), minimizing the size of the robots body and facilitating changes of the articulations. AZIMUT also adopts a distributed embedded approach. Low-level control of the platform is done using distributed subsystems (for motor control, sensing, energy distribution, etc.), each equipped with its own microcontroller board (PIC, Infineon, Motorola, etc., determined according to the computing requirements). Communication between the microcontrollers is done using Control Area Network protocol (CAN 2.0B) on a 1 Mbps shared bus, and our own 92 bits frame protocol. Improved robustness and flexibility is therefore provided, since microcontroller boards can be easily switched or replaced in case of problems. It also facilitates debugging and subsequent designs and extensions of the platform.

Figure 4 presents AZIMUT second prototype, with its wheel articulations. This prototype led to the design of a wheel-motor actuator (made with a brushless motor coupled with a harmonic drive reducer), providing propulsion to the robot in a wheel configuration or a track configuration. This actuator is lighter and more powerful than the one used in the first prototype. AZIMUT-2 is smaller (60 cm×52 cm×29 cm compared to 71 cm×53 cm×38 cm), lighter (33.7 kg

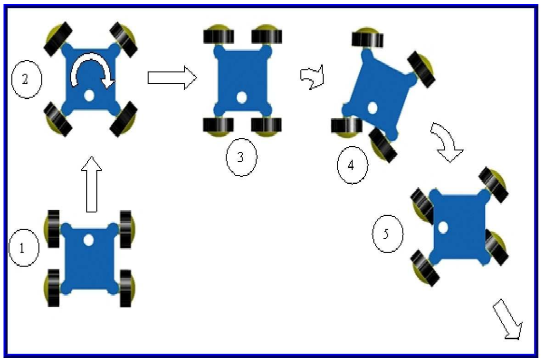


Figure 3: Possible motion configurations with AZIMUT.

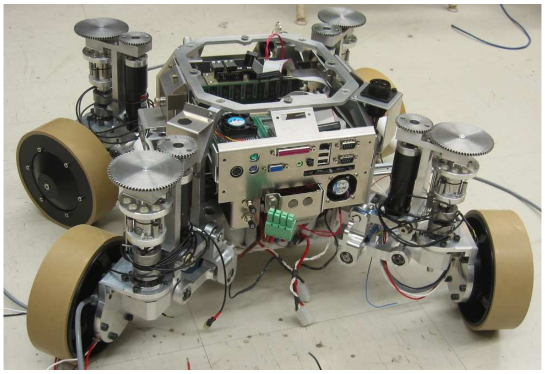


Figure 4: AZIMUT-2.

Figure 5: Spartacus (front view, back view).

vs 63.5 kg) and faster (directional speed of 180°/sec vs 120°/sec, no load; maximum velocity of 1.4 m/s vs 1.2 m/s) than the first prototype. The prototype also increases stability and compliance by adding a vertical suspension and by using elastic actuators for the motorized direction of AZIMUTs articulations. An elastic element (e.g., a spring) is placed in the actuation mechanism and a sensor is used to measure its deformation, allowing to sense and control the torque at the actuators end (Pratt & Williamson 1995; Robinson 2000; Zinn *et al.* 2004). This should improve robot motion over uneven terrains, making the robot capable of “feeling” the surface on which it operates. Such actuators are usually exploited in robotic manipulators, and their use on a mobile platform (wheeled or tracked) is novel. AZIMUT-2’s electrical and computing distributed embedded system also has been improved. Small, stable and robust CAN modules (Direct Current (DC) Brush/Brushless 10 Amps motor drive; energy distribution; sonar sensor interface; LCD 128×64 bits monitor; wireless remote control) and I2C modules (DC 3 amps motor drive; infrared range sensor interface for stand-alone sensors). A Microchip PIC18F microcontroller board has been designed (and named the PICoModul). We also have developed an innovative and efficient way to remotely program PICoModul over the CAN bus, allowing to easily monitor and change their configuration online, facilitating also maintenance. Visualizing state variables (values, graphs of temporal evolution) using our graphical interface greatly simplifies calibration of the robot.

Spartacus, Interactive Capabilities for Mobile Robots

Spartacus is the robotic platform we have designed for high-level interaction with people in real life settings. The robot built is shown in Figure 5. It is a differential steering wheeled platform with a humanoid shape. Spartacus carries different kinds of interfaces, such as audio amplifier and speakers, a LEDs electronic panel on its back, a touchscreen and a business card dispenser on its front. The robot also carries its own chargers so that it can be plugged directly into a regular electric outlet. To facilitate the integration of all the required components, our design follows a distributed approach. Different subsystems communicate with each other to exchange information and coordinate their actions (Michaud *et al.* 2005b), just like AZIMUT and Tito.

Spartacus is also equipped with a SICK LMS200 laser range finder (for autonomous navigation), Sony SNC-RZ30N 25X pan-tilt-zoom color camera, an array of eight microphones placed in the robot’s body. High-level processing is done using an embedded Mini-ITX computer (Pentium M 1.7 GHz). The Mini-ITX computer is connected to the low-level controllers through a CAN bus device, the laser range finder through a serial port, the camera through a 100Mbps Ethernet link and the audio amplifier and speakers using the audio output port. A laptop computer (Pentium M 1.6 GHz) is also installed on the platform and is equipped with a RME Hammerfal DSP Multiface sound card using eight analog inputs to simultaneously sample signals coming from the microphone array. Communication between the two on-board computers is accomplished with a 100Mbps Ethernet link. Communication with external computers can be accomplished using the 802.11g wireless technology, giving the ability to easily add remote processing power or capabilities if required.

Numerous algorithms has been integrated to provide Spartacus with interactive capabilities. MARIE (Mobile and Autonomous Robot Integrated Environment)[1] is our middleware framework allowing to link multiple software packages. Spartacus' implementation integrate the following robotic software tools:

- RobotFlow and FlowDesigner (FD) (Cote *et al.* 2004), two modular data-flow programming environments that facilitates visualization and understanding of the robots control loops, sensors, actuators. These environments are used to implement the behavior-producing modules, the vision modules for reading messages (Letourneau, Michaud, & Valin 2004) and DECIBEL, our real-time sound source localization, tracking (Valin *et al.* 1) and separation (Valin, Rouat, & Michaud 2004) system.
- ConfPlan (Beaudry *et al.* 2005) is the approach used for task planning with metrics and temporal constraints. It is implemented in C++.
- Carmen (Carnegie Mellon Robot Navigation Toolkit) for path planning and localization (Montemerlo, Roy, & Thrun 2003).
- Player for sensor and actuator abstraction layer (Vaughan, Gerkey, & Howard 2003), Stage/Gazebo for 2D and 3D simulators, and Pmap library[2] for 2D mapping, all designed at the University of Southern California.
- Nuance[3] for speech recognition and Festival for speech generation.
- OpenCV library for vision algorithms.
- QT3 for the graphical interface development and libXML2 for configuration files and data representation language.

Motivated Behavioral Architecture (MBA)

For autonomous use of all these capabilities, we are developing a computational architecture based on the notion of motivated selection of behavior-producing modules (BPM). The architecture, referred to as Motivated Behavioral Architecture (MBA), contains different motivational sources derived from perceptual influences, pre-determined scenarios, navigation algorithms, a planning algorithm or other types of reasoning algorithms. One reason for distinguishing these influences through different motivational sources is to simplify programming of the robot's intentions in accomplishing various tasks.

Figure 6 represents the implemented architecture. It is composed of three principal elements. BPMs define how particular percepts and conditions influence the control of the robot's actuators. They can be typical behavior-based reactive controllers with or without internal states, goal-oriented behaviors or other types of behaviors. The actual use of a BPM is determined by the arbitration scheme and the BPM's activation conditions, as derived by the BPM Selection module. BPM Arbitration scheme used in this implementation is priority-based. The BPM Selection module determines which BPM are to be activated according to recommendations (*rec*) made by motivational sources (or Motivations) concerning tasks. Recommendations can either be negative, neutral or positive, or take on real values within this range. The activation values (BPM.Activation) reflect the robot's combined intentions derived from interactions between the motivational sources.

Spartacus' motivational sources are categorized as either instinctual or rational. Instinctual motivations provide basic operation of the robot using simple rules. Rational motivations are more related to cognitive processes, such as navigation and planning. Motivations can be derived from percepts, from results and states of current goals, and from monitoring tasks. One additional source of information for motivations is the observation of the effective use of the behaviors, represented by the link BPM.Exploitation. Such information can serve as an abstraction of the robot's interactions within the environment. An active behavior may or may not be used to control the robot depending on to the sensory conditions it monitors and the arbitration mechanism used to coordinate the robot's behaviors. So, an active behavior is exploited only when it provides commands that actually control the robot. All the MBA's modules run concurrently to derive goals and expressing behavior recommendations.

To facilitate addition of new capabilities and insure independence from the robotic platform, motivations are kept as generic and independent from each other and from the BPM as much as possible, through the use of tasks. This is managed by the Dynamic Task Workspace (DTW) and System Know How (SNOW) modules. The DTW organizes tasks in a hierarchy using a tree-like structure, from high-level/abstract tasks (e.g., deliver message), to primitive/BPM-related tasks (e.g., avoid obstacles). Through DTW, motivations share knowledge about how to activate, configure and monitor BPM. Motivations can add and modify tasks (by submitting modification requests m), request information about them (q) or subscribe to events (e) regarding the task's status. The SNOW module defines and communicates tasks parameters (p) and behaviors results (*res*) between BPM and DTW.

Our implementation attempted to have Spartacus participate in all the 2005 AAAI Mobile Robot events. This choice ensures that the integration work is not oriented toward solving one particular application or task. Our objective in 2005 is to validate MBA's working principles in a complete working system, integrating specific interaction capabilities that we want to demonstrate with our 2005 entry as follows:

- **Navigation**. Pmap and Carmen libraries are now made available through MARIE, allowing the production of a map (using Pmap) that can be used by Carmen for autonomous navigation. To use Carmen, laser range data must be taken as the robot is exploring an area (whilst being controlled), and this data is then analyzed off-line us-

---

[1] http://marie.sourceforge.net

[2] http://robotics.usc.edu/ ahoward/pmap/

[3] http://www.nuance.com/

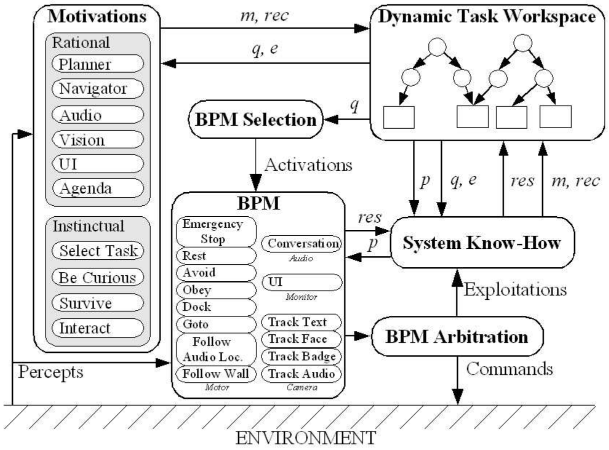


Figure 6: Spartacus' computational architecture.

ing a mapping algorithm. Also, manual corrections to the map may sometimes be required to ensure safe navigation of the robot. Pmap facilitates the generation of maps from laser range data, and combining it with Carmen will allow building maps on-line while using Carmen for navigation and path planning.

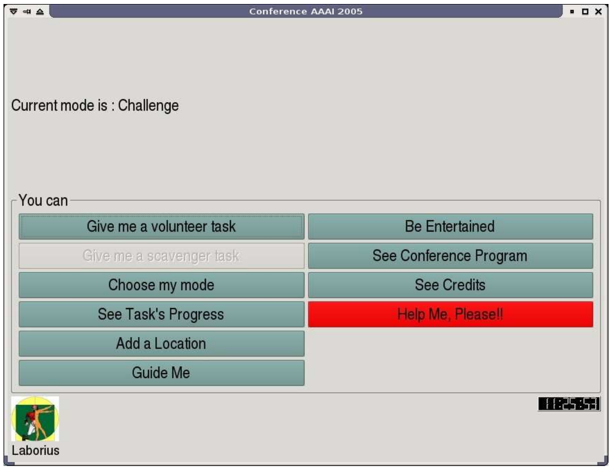
- **Vision Processing.** We previously had developed an algorithm that can extract symbols and text from a single color image in real world conditions (Letourneau, Michaud, & Valin 2004). The algorithm has been trained to recognize Arial fonts with characters having a minimum height of 40 pixels. The zoom capability of the camera allows the camera to 'zoom in' on the regions containing text and to adjust the image resolution so that it can successfully recognize each character. Using a 12X optical zoom camera, the system is able to read printed text on 8.5" × 11" sheet of paper with font size 128 at a maximum distance of 4.5 meters. With the 25X optical zoom camera installed on Spartacus, such text can be read at distances as far as 10 meters. For conference name tags using smaller font size (12-18), recognition is possible at a maximum distance of 30 cm with a 12X optical zoom, and at 2.5 meters with the 25X zoom.

  We also integrate an object recognition algorithm (Lienhart & Maydt 2002) that can identity regions of interest in the image such as human faces and silhouettes. Once identified, these regions can be tracked using color information, as achieved in (Perez *et al.* 2002). Combining object recognition and tracking algorithms can reduce the processing required when detecting regions of interest. Also while tracking a region, it is possible to adapt the region models to the current observed conditions in a dynamic environment. Thus, a person can be identified when facing the camera by the recognition algorithm. Once this is achieved, the tracking algorithm would attempt to maintain an appropriate color model of the person. This would enable the algorithm to keep track of the person, even if the face changes direction or the lighting conditions change.

- **Sound Processing.** Spartacus' artificial audition system uses an array of eight microphones. Our approach is capable of simultaneously localizing and tracking as up to four sound sources that are in motion over a 7 meter range, in the presence of noise and reverberation (Valin *et al.* 1). We have also developed a method to separate in real time the sound sources (Valin, Rouat, & Michaud 2004) in order to process communicated information from different interlocutors using software packages such as Nuance. We tested Nuance with data generated by our system using speech utterances (i.e., four connected digits spoken in English) simultaneously made by three separate speakers. Preliminary recognition performance observed with Nuance is 84% for the three sources combined. For just one source located in front, the observed performance is 87%, and only one out of five persons was able to reproduce the same performance (the average was 81%).

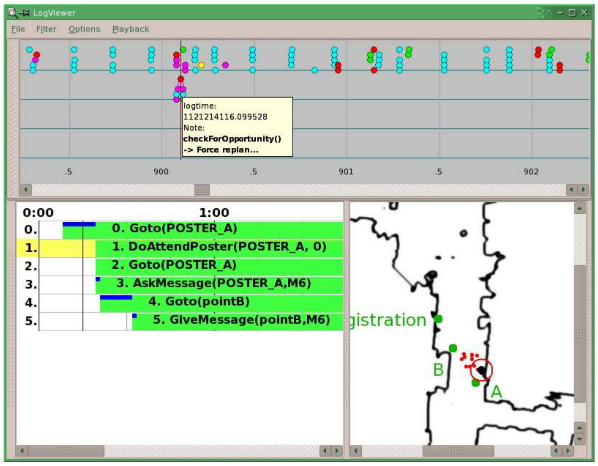
Four actuators are controlled by BPMs, each of them having one or more associated behaviors. The motor actuator has eight associated behaviors: Emergency Stop, stopping the robot when abnormal conditions are detected; Rest, stopping the robot from moving; Avoid, making the robot move safely in the environment; Obey, following user orders; Dock, stopping the robot while waiting to be connected to a charging station; Goto, directing the robot to a specific location; Follow Audio Localization, making the robot follow an audio source; Follow Wall, making the robot follow a wall (or corridor) when detected, otherwise generating a constant forward velocity. The Camera actuator is controlled by four behaviors: Track Text; Track Face; Track Badge; Track Audio, pointing the camera toward a audio source detected. The audio actuator (i.e., the sound card output) is associated with the Conversation behavior. The Monitor actuator is associated with UserInterface (UI) behavior, using the monitor to interact with users.

In this implementation, rational motivations has a greater priority in case of conflicts with other ones. For instinctual motivations, Select Task selects one high-level task when none has yet been prioritized. For instance, between tasks that require the robot to go to a specific location, this motivation selects the task where the location is physically closest to the robot. Be Curious motivates the robot to discover and collects information on its environment. Survive urges the robot to maintain its physical integrity by recommending to avoid obstacles. Interact is a process allowing the robot to socialize with people. The other modules are for rational motivations. Planner is where our planner can influence the decisions of the robot. In MBA, the role of the planner is to provide the robot with the capability of determining which primitive tasks and which sequence of these tasks are necessary to accomplish high-level tasks under temporal constraints and limited capabilities (as defined by the set of BPM). Our first implementation is a simple reactive planning module that interleaves planning and executing (Beaudry *et al.* 2005), like (Haigh & Veloso 1998) and (Lemai & Ingrand 2004). Navigate determines the path to a specific location, as required for tasks in the DTW.



Current mode is : Challenge

Figure 7: Touchscreen interface.



logtime:
1121214116.099528
Note:
checkForOpportunity()
-> Force replan...

0. Goto(POSTER_A)
1. DoAttendPoster(POSTER_A, 0)
2. Goto(POSTER_A)
3. AskMessage(POSTER_A,M6)
4. Goto(pointB)
5. GiveMessage(pointB,M6)

Figure 8: LogViewer (partial).

Audio and Vision motivate the robot to do tasks according to their respective senses (e.g., track badge, localize sound source). UI is a process allowing the execution of user's requests for tasks. Agenda generates predetermined tasks to accomplish according to the AAAI Mobile Robot Competition context.

Results

Spartacus did entered all events (Challenge, Scavenger Hunt, Open Interaction) at the 2005 AAAI Mobile Robot Competition, using the same implementation. Each event was programmed as a special configuration mode, which could be requested by the experimenter (using speech or from the tactile interface, as shown in Figure 7). Complete integration was demonstrated, combining map building with localization and path planning, audio tracking of speakers around the robot (positioning the camera in the direction of where the sounds came), speech recognition in "cocktail party effect" conditions, scripted vocal response to specific requests (around 30 different requests), reading the room names (printed on 8.5" × 11" sheets of paper, printed in Arial font and not in Times, which was the font use for the badges), and dynamic scheduling of tasks set according to temporal constraints. Events such as recognizing a specific sentence or reading a particular message could be memorized and tagged on the map for further reference.

This first implementation also allowed us to identify different types of limitations. The first set of limitations are caused by the experimental conditions being extremely challenging. Spartacus' motor drives were set to a secure speed when running on hard surface conditions, and not on carpet (making the robot go slower). Redirecting toward one of the five possible elevators in the hotel, as the doors open up, Spartacus did not have enough time to get onboard before the doors would close. The trials were also conducted in unrealistic conditions (registering to a conference does not usually happen right in the middle of a reception, when tables are being installed, in conditions in which it is even difficult for humans to sustain a conversation, and for short periods of time). To handle complex situations that may arise when completing the events such as the Challenge from start to end, it requires running the robot for 1 to 1.5 hour (which is approximately the energetic capability of the robot). Trying to demonstrate the capabilities of a planner is therefore difficult in limited periods of time. Another observation is that Spartacus should be more directive in what it wants to do (e.g., asking people to move away if it has to do something), and not always keep on responding to people's requests. Also, with a robot that can hear in such difficult conditions, it can also understand when someone is trying to explaining out loud what the robot is doing (during demos for instance), which may interfere with what it is supposed to be doing. Spartacus only stops listening when it speaks so that it does not try to understand itself. This limits the amount of unnecessary processing, but does not allow the robot to understand a request made by someone as it speaks.

Second, with many capabilities integrated on the same platform, it becomes difficult to recognize what the robot is actually doing. One tool that revealed to be very useful is our LogViewer, partially shown in Figure 8. The upper section of the LogViewer contains a timeline view of DTW (first line) and planner events (second lines), and behaviors' activations and exploitations (not shown). The bottom section shows detailed information according to the position of the horizontal marker on the timeline: a list of DTW's tasks and properties (not shown), active motivations (not shown), the current plan (shown), the map of the environment (shown) and the trajectory of the robot (not shown), the behaviors and their activations and exploitations (not shown). Another useful tool would be to have a GUI interface to configure and manage the different software components executed on multiple processors through MARIE.

Third, optimization is still required to make the integration work smoothly. One improvement would be to clearly display what the robot is currently doing on its touchscreen interface or its back LDC electronic panel. Spartacus sometimes stopped, not listening to vocal requests, because it was trying to read an unanticipated but potential message in its view. Displaying what the robot said revealed quite useful, as it sometimes was difficult to hear what the robot was saying because of the high-level of noise conditions in the environment. The interface could be improved by displaying bigger graphical object and messages. Speaker volume could not be set to its maximum since a humming sound coming out of the speakers is interfering with two of the nearby microphones. Our algorithm for reading messages needs fine-tuning and maybe the use of a more sophisticated optical character recognition algorithm to avoid false detections. To demonstrate Spartacus' speech recognition capabilities in open settings, we had to sacrifice visual tracking of people. More CPU processing power is required to fully integrate all of our available vision software. Quantifying performances limitations in our implementation, for audio and vision processing, are also required.

Related Work

Our first attempt to completing the entire AAAI Challenge was in 2000. Our entry, a Pioneer 2 robot named Lolitta, used sonars as proximity sensors, navigated in the environment by reading written letters and symbols, interacted with people through a touch-screen interface, displayed a graphical face to express the robot's emotional state, determined what to do next using a finite-state machine, recharged itself when needed, and generated a HTML report of its experience (Michaud *et al.* 2001). EMIB (Emotion and Motivation for Intentional selection and configuration of Behavior-producing modules) was the computational architecture used in this implementation (Michaud 2002). MBA is a generalization of EMIB in which motivational sources are grouped and exchange information asynchronously through the DTW. The SNOW module, not present in EMIB, separates task representation from BPM. Compared to Lolitta, the additions made to Spartacus provide a more flexible environment allowing the expansion and integration of further capabilities over time.

The two other autonomous robotic entries that attempted to complete the Challenge are Lewis and Grace. Lewis (Smart *et al.* 2003) is a B21 robot platform equipped with a pan-tilt-zoom color camera and a laser range finder. As with Lolitta, Lewis was able to recognize symbols (arrows) enabling navigation in the world, interacted with people using a touchscreen interface, and used a finite-state machine to go through the different steps of the challenge. Its novel feature was its framing algorithm, allowing it to take pictures of people in open settings (Byers *et al.* 2003). Grace (Simmons *et al.* 2003) is also a B21 robot base equipped with a 15 inches display (showing a virtual face), a laser range finder, a pan-tilt-zoom color camera, a pan-tilt stereo vision head and one wireless microphone. Instead of using symbols, directions to reach the registration desk were provided by an operator using vocal commands. Speech recognition however appeared to be difficult because of background noise. The registration desk was detected by localizing a big pink poster board. Grace used its laser range finder to detect the waiting line for registration. Once at the registration desk, Grace interacted vocally with conference personnel. Autonomous navigation was then possible using the laser and a pre-generated map. One of the biggest challenges faced was the integration of the different computational modules, which were manually initiated at the appropriate time during the different steps of the challenge. No global computational architecture seems to have been used.

Finally, one other related work is HERMES (Bischoff & Graefe 2002). Like Spartacus, it is a directional steered wheeled platform with a humanoid shape. This base has two manipulator arms, features that Spartacus does not have. HERMES combines visual, kinesthetic, tactile and auditory sensing with natural spoken language (input via voice, keyboard or e-mail). Robot control is achieved using a situation-oriented skill- and behavior-based paradigm, with a central situation recognition module. HERMES' work concentrates more on situation-dependent natural language communication and kinesthetic sense. Our focus is on enhanced artificial audition, different interaction modalities using vision and graphical displays, all integrated into a common architectural methodology involving autonomous, high-level reasoning (such as planning).

Conclusion

With this paper, our objective is to demonstrate that it is possible to design multiple types of socially interactive robots, addressing more than just programming these devices. By doing so, we want to outline the challenges that we must face to eventually have such machines work in real life settings. Overall, designing a mobile robot that must operate in public environments probably addresses the most complete set of issues related to interactive systems. The process becomes complex and takes time, with system integration playing a fundamental role at all levels. It also outlines the challenge of evaluating visual, auditive and graphical interfaces in creating sophisticated human-robot interaction. Combining a variety of capabilities increases the complexity and diversity of interaction the robot can have. This in turn leads to greater complexity and diversity of analyzing and understanding such increased capabilities, leading to the rich field of interaction studies.

Our work is still in progress for completing the design, but already important benefits are observed. For instance, working with an integrated programming environment such as MARIE allows us to focus on the decision-making issues rather than on software programming considerations. Frameworks like MARIE are more than just engineering tools: they are part of the scientific process of studying and designing autonomous systems. We hope with MARIE to contribute to such effort, and have others join the initiative for continued progress leading to new discoveries. Also, along with the MBA architecture and the Spartacus robotic platform, MARIE provides a common framework to test and to compare different algorithms such as navigation tools and planning algorithms. We believe that the AAAI Mobile Robot events could become a nice venue to conduct such studies, extending its goals of demonstrating applied artificial intelligence research to the public. The objective would be to hosting standardized experimental trials from which scientific research could be gathered, sharing algorithms and benefiting from their integration. This would contribute in making the event even more interesting, directly taking on the long-term goal of artificial intelligence.

Acknowledgments

F. Michaud holds the Canada Research Chair (CRC) in Mobile Robotics and Autonomous Intelligent Systems. Support for this work is provided by the Natural Sciences and Engineering Research Council of Canada, the Canada Research Chair program and the Canadian Foundation for Innovation.

References

Beaudry, E.; Brosseau, Y.; Cote, C.; Raievsky, C.; Letourneau, D.; Kabanza, F.; and Michaud, F. 2005. Reactive planning in a motivated behavioral architecture. In *Proc. American Association for Artificial Intelligence Conf.*.

Bischoff, R., and Graefe, V. 2002. Dependable multimodal communication and interaction with robotic assistants. In *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication*.

Byers, Z.; Dixon, M.; Smart, W. D.; and Grimm, C. M. 2003. Say cheese! experiences with a robot photographer. In *Proc. Innovative Applications of Artificial Intelligence Conf.*.

Cote, C.; Letourneau, D.; Michaud, F.; Valin, J.-M.; Brosseau, Y.; Raievsky, C.; Lemay, M.; and Tran, V. 2004. Code reusability tools for programming mobile robots. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

Duquette, A. 2005. L'exploration d'une intervention avec un robot-jouet anime chez les enfants autistes. Master's thesis, Universite de Sherbrooke, Departement of Psycho-Education.

Haigh, K., and Veloso, M. 1998. Planning, execution and learning in a robotic agent. In *Proc. Int. Conf. on Artificial Intelligence Planning Systems*, 120–127.

Lemai, S., and Ingrand, F. 2004. Interleaving temporeal planning and execution in robotics domains. In *Proc. National Conf. on Artificial Intelligence*.

Letourneau, D.; Michaud, F.; and Valin, J.-M. 2004. Autonomous robot that can read. *EURASIP Journal on Applied Signal Processing* 17:1–14.

Lienhart, R., and Maydt, J. 2002. An extended set of haar-like features for rapid object detection. In *Proc. Int. Conf. on Image Processing*.

Michaud, F., and Caron, S. 2000. Roball – an autonomous toy-rolling robot. In *Proc. Workshop on Interactive Robotics and Entertainment*.

Michaud, F.; Audet, J.; Letourneau, D.; Lussier, L.; Theberge-Turmel, C.; and Caron, S. 2001. Experiences with an autonomous robot attending the AAAI conference. *IEEE Intelligent Systems* 16(5):23–29.

Michaud, F.; Laplante, J.; Larouche, H.; Duquette, A.; Caron, S.; Letourneau, D.; and Masson, P. 2005a. Autonomous spherical mobile robotic to study child development. *IEEE Trans. on Systems, Man, and Cybernetics* 35(4):1–10.

Michaud, F.; Letourneau, D.; Arsenault, M.; Bergeron, Y.; Cadrin, R.; Gagnon, F.; Legault, M.-A.; Millette, M.; Pare, J.-F.; Tremblay, M.-C.; Lepage, P.; Morin, Y.; and Caron, S. 2005b. Multi-modal locomotion robotic platform using leg-track-wheel articulations. *Autonomous Robots* 18(2).

Michaud, F. 2002. Emib – computational architecture based on emotion and motivation for intentional selection and configuration of behaviour-producing modules. *Cognitive Science Quaterly* 3-4:340–361.

Montemerlo, M.; Roy, N.; and Thrun, S. 2003. Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (Carmen) toolkit. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2436–2441.

Perez, P.; Hue, C.; Vermaak, J.; and Gangnet, M. 2002. Color-based probabilistic tracking. In *Proc. European Conf. on Computer Vision*, 661675.

Pratt, G., and Williamson, M. 1995. Series elastic actuators. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, 399–406.

Robins, B.; Dautenhahn, K.; te Boekhorst, R.; and Billard, A. 2004. Robots as assistive technology – does appearance matter? In *Proc. IEEE Int. Workshop on Robot and Human Interactive Communication*, 277–282.

Robinson, D. W. 2000. *Design and analysis of series elasticity in closed loop actuator force control*. Ph.D. Dissertation, MIT.

Salter, T.; Michaud, F.; Dautenhahn, K.; Letourneau, D.; and Caron, S. 2005. Recognizing interaction from a robot's perspective. In *Proc. IEEE Int. Workshop on Robot and Human Interaction Communication*.

Simmons, R.; Goldberg, D.; Goode, A.; Montemerlo, M.; Roy, N.; et al. 2003. Grace : An autonomous robot for the AAAI robot challenge. *AI Magazine* 24(2):51–72.

Smart, W. D.; Dixon, M.; Melchior, N.; Tucek, J.; and Srinivas, A. 2003. Lewis the graduate student: An entry in the AAAI robot challenge. Technical report, AAAI Workshop on Mobile Robot Competition.

Valin, J.-M.; Michaud, F.; Hadjou, B.; and Rouat, J. 1. Localization of simultaneous moving sound sources for mobile robot using a frequency-domaine steered beamformer approach. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1033–1038.

Valin, J.-M.; Rouat, J.; and Michaud, F. 2004. Enhanced robot audition based on microphone array source separation with post-filter. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

Vaughan, R. T.; Gerkey, B. P.; and Howard, A. 2003. On device abstractions for portable, reusable robot code. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2421–2427.

Zinn, M.; Kathib, O.; Roth, B.; and Salisbury, K. 2004. A new actuation concept for human-friendly robot design: Playing it safe. *IEEE Robotics and Automation Magazine* 11(2):12–21.