Architectural Aspects of Networked Robotic Systems

João Sequeira
Instituto Superior Técnico / Institute for Systems and Robotics
Av. Rovisco Pais 1
1049-001 Lisbon, Portugal
jseq@isr.ist.utl.pt

Abstract — This paper presents a simplified architectural description for the urban surveillance experiment under development within the URUS project.

The description contains two views in the computational viewpoint that are also applicable to other classes of network robotics problems. These are a functional layer through which the system has basis operational capabilities, and a human layer that concentrates the systems related to human-robot interaction.

Keywords: *Networked Robots, Robot Control Architecture, Architectural Description*

I. Introduction

Since mid 90s that the dissemination of low cost networking media is enlarging the robotics applications domain. In fact, networking technology is currently an economical and societal driving force that is pushing forward the integration in society of a number of other technologies such as robotics. Potential improvements in the quality of life of the citizens easily point to high revenue markets. As an example, surveillance systems are increasingly relevant in a variety of domains. Networked robotic surveillance systems, operating in cooperation with human supervisors and additional fixed sensing devices, can cut down the operation costs while maintaining or increasing operational standards.

Different network robotics applications share the underlying technologies and functionalities. The harsh environments often found in search and rescue applications and the regular urban settings require similar sensing, automated task decomposition schemes, and decision making functionalities Robot(namely distributed) or, more generally, ubiquitous computing and ambient intelligence functionalities (see [6]).

The URUS project aims at demonstrating the capabilities of robots interacting with humans, smart devices, and with other robots, connected as a network, in surveillance and assistance to people in urban environments, [18]. The networking media is the internet, both in wireless and cable forms. Two teams of heterogeneous robots will be deployed in two sites in the city of Barcelona, Spain. The surveillance application, considered in this paper, focus on (i) detecting emergency conditions that require the evacuation of people out of the test site, and (ii) guiding people through the corresponding escape routes with the help of a team of three to five robots. The interaction between the robots and humans is to be natural, in the sense that no special a priori knowledge by the people is to be required for the interaction with the robots. The surveillance application will be tested at the north campus of Universidad Politecnica de Catalunya (see Figure 1 for a set of views) in a square area of approximately 100 meters long side length. Around 20 IP cameras are placed outside the buildings to cover most of the test area. These cameras provide the information for detecting emergency conditions, people activities, and robot localization, among other functionalities. The photos show a regular surface that poses no special locomotion problems to common wheeled robots.

An emergency situation can be declared by people using standard interfacing devices such as mobile phones running specific applications or simply by waving to the fixed cameras installed outdoors (a typical human sign to call for attention). Once the emergency location is computed the robots are sent to the place to guide the people. While converging to the emergency location the robots warn people to move away from unauthorized routes. Upon arrival the robots try to confine the people to the interior of a formation and move towards the escape point maintaining this formation.

The architectural goals of the URUS system are common to most of networked robotic systems (NRS) independently of the specific application envisaged, (i) all the services required by the specific task envisaged (sensors, motor schemas, communications, generic queries), (ii) protection against the use by unauthorized entities, (iii) flexibility and scalability, and (iv) network transparency in software development, (v) promote software reuse and programming language independence, and (vi) platform independence. The discussion in this paper is just a small part of the ongoing effort on the design of an architecture to integrate all the components and meet the aforementioned goals. Results on specific components of the project, such as the image processing and robot prototype construction can easily be found through the publications of the project and at the its website (www-iri.upc.es/groups/urus/).

The paper is organized as follows. Section II briefly points to a number of NRS architectures and concepts from which one can draw inspiration. Section III presents a simplified architectural description being assessed within the URUS context. Section IV concludes the paper.

(a) (b) (c)

Fig. 1. A set of views of the test site for the surveillance experiment at the UPC north campus

II. Architectures for robotic systems

Multiple architectures for robotic systems have been proposed in the literature. The similarity between architectures in all domains has been recognized by some authors, [4], and constitutes a useful conjecture when surveying the field. The natural consequence of this conjecture is that a wide variety of architectures can be used in applications involving NRS, provided that the network becomes transparent to the components. A side effect of this conjecture is that innovation in the field of architectures tends to grow slowly.

The NIST-RCS architecture, [4], has been referenced as being suitable to intelligent systems. It comprises sensor, actuator, perception, world modeling, value judgement and behavior generation blocks. The perception, value judgement and behavior generation blocks can query the world model to decide on what to do. Once the implementation abstracts the network, each of these blocks can be distributed to form the nodes in the network.

The DAMN architecture, [11], is a collection of independently operating modules implementing a group of distributed behaviors. These communicate with a centralized arbiter which is responsible for combining the behaviors such that the resulting action reflects their objectives and priorities.

The CAMPOUT architecture for multiple robots, [8], focus on a hierarchy of behaviors. The key components of this architecture include device drivers and communication layers in addition to primitive behaviors and behavior composition strategies.

The nodes in the network proposed in [19] are composed by a robot and an interface to a network bus. The network interface contains an application interface, a query engine, a result collector and a context database and manager. The query engine broadcasts all the interfaces in the network for the contexts referred in the query. Such broadcasts are received and processed by a context data manager, put in the network, and grabbed by destination query result collector and then forwarded to the application that originated the request.

A three tier architecture with an application layer, and infrastructure services layer and a middleware layer is proposed in [7]. The application layer contains the functional blocks related to single robot activities, e.g., path planning. The infrastructure layer provides the network services. The middleware layer handles the communication between services.

The KAMRO architecture for distributed robots, [25], is a standard network with each node containing communications, knowledge base, planning and device dependent subsystems.

The Distributed Field Robot Architecture (DFRA), [14], is a behavioral architecture that implements a standard perception-to-actuation conceptual scheme, with additional blocks for map building and sensor and actuator management. The DFRA adds two layers for distributed resource protection and abstract representation of single robots over the network.

A five layer architecture, with device managing, control, network, integrating and application layers, for distributed robot systems is described in [24]. The device managing layer interfaces the physical devices with the rest of the architecture and contains the low level control strategies. The control layer acquires high-level data, e.g., face and voice recognition and navigation data. The network layer hides the communication interfaces among the modules. The integrating layer contains a map manager, a global planner and a database system. The application layer is used by to define the tasks for the overall system using abstract commands.

The MRHA architecture, [17], used in indoors security patrolling, is based on the OSI/ISO reference model. It includes seven layers, application, presentation, session, transport, network, data-link and physical. The mobile platforms are seen as resources, controlled from a set of networked computers. The network includes a supervisor computer, distributed databases, planning/dispatching computer and a communications computer. The supervisor manages the overall system, including the operator stations. The database computers estimate the positions of the robots over time using the raw data acquired from the robot. The planner/dispatcher computers are in charge of navigation and collision avoidance. The communications computer handles the wireless communications with the mobile platforms

The DARPA Software for Distributed Robotics, [12], addressed the distributed networking of large size (100 plus) robot teams operating in indoor missions. This architecture is supported on the Saphira/Aria software. A distributed dispatcher manager organizes the team as a hierarchy of roles.

Optimal distributed map building, fault tolerant communications are just two of the functionalities available to each robot.

The last examples in this brief state of the art emphasize the role of networking technologies. A CORBA based architecture with two layers, infrastructure and service, is described in [27]. The infrastructure contains the classes that support the service layer but can also be used by user defined applications. The service layer provides the high-level services the users can deploy in their applications.

An object oriented architecture to control a distributed surgical robotics system is described in [16]. This architecture also uses the CORBA framework to implement object distribution. The overall system includes multiple computing hosts running on different operating systems and software languages.

The Miro framework, [23], is a distributed object oriented framework for robot control also supported on CORBA. Miro represents a middleware layer for autonomous robots, providing network transparency, event based publisher, logging facilities, and sensor and actuator services.

Additionally, the effort by the robotics community in architecture design is being disseminated through the RoSta project, [1].

III. A SIMPLIFIED ARCHITECTURAL DESCRIPTION

The architecture in this paper gets inspiration from multiple works described in the NRS literature and on the principles used in the development of software intensive projects. The IEEE 1471 methodology, [9], establishes an ontology for architectures, applicable to a wide range of software development projects and directly applicable to robotics. For open distributed processing systems, five *viewpoints* are usually considered, namely, enterprise, information, computational, engineering and technology, [9].

Using the IEEE 1471 parlance, this paper describes two key *views* that implement the computational *viewpoint*. These describe (i) how the system executes a task successfully accounting only for a minimal human-robot interaction directly issued by system supervisors, and (ii) how the system interacts with humans in a "natural" way.

The first *view*, named *functional*, corresponds to a basic approach to NRS commonly found in the literature. In this *view*, the system is a network of functional components and it covers, amongst others, the computational objects, interactions, and interfaces.

The second *view*, named *human*, adds to the system the ability to interact with people and have its behavior changed as a result of the interactions.

These two *views* are hierarchically dependent. The functional view is identified with a base layer that ensures a minimal operational performance level, whereas the human view adds the ability of the system to interact with humans in a natural way and adjust its behavior according to measures of human emotional response. A motivation for this structure can be drawn from Personality Models in Psychology (see for instance [13] on the hierarchy of needs model). The functional view provides the basic survival skills whereas the human view provides the social skills lying above survival.

A. *The functional* view

Generically, under reasonable assumptions of the availability of adequate sensor data and adequate task design, an autonomous NRS can execute a mission with minimal supervision. The NRS responds to the stimulae by the environment and adjusts its behavior accordingly using the resources available on the network. The functional *view* describes the organization of these resources. The main physical devices involved are shown in Figure 2. Sensors include internet cameras, eventually with some local image processing facilities, and MICA sensor boards for generic purposes. A Central Station (CS) provides generic computational facilities that can be used by any of the other devices. The task planning and allocation services and the system management services run at this component. Though from a conceptual point of view the CS is a single entity, it is worth to remark that it can be implemented as a distributed set of components.

The entities in the network are providers of resources (or services) to the whole system and hence the overall system naturally becomes a service oriented architecture (SOA)[1]. Services can be used to wrap specialized blocks of code developed by the project team so that it can be accessed from other services, for instance located in a remote platform. The way services interact with each other and with the rest of the environment is specified through *service interfaces* (see the Open Service Interface Definitions, [3], and OASIS, [2]), providing, for example, request-reply, and access to service internals. The SOA concept allows that users requesting a service only need to know the rules for the use of the specific interfaces avoiding dealing with internal details of the service itself. Describing the particular set of services available within URUS can be made by *views* in the computational and engineering *viewpoints* and is outside the scope of this paper.

Figure 3 describes the components in each robot. The COMM component handles everything that is related to communications between the robot and the outside environment. The GNC component handles the guidance and navigation of the robot. The Local Task Manager controls the execution of the assigned tasks. The CS uses WiFi and LAN/WAN interfaces to exchange data with robots and sensors. The URUS system can also be accessed directly through a cellular phone interface. In this case the CS receives and processes the requests, eventually requiring the robots to execute some task. In addition, robots have Bluetooth interfaces that can be used by the applications in the cellphone to contact the URUS services without requiring a normal phone call to be made.

Although the functional view only targets limited interaction capabilities, the main actors must include a form of evaluating the effect of their actions on the environment and adjust future actions to reflect this evaluation. Figure 4 shows the performance monitoring loops involved in the URUS surveillance

[1]SOA are often identified with networks of distributed computing modules, each with methods that control its own operating dynamics.

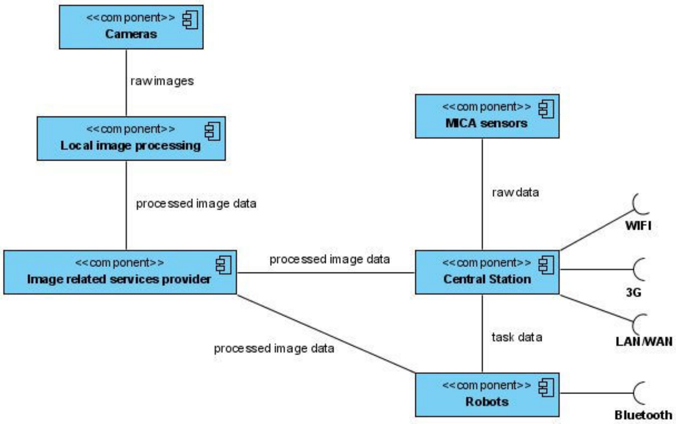

Fig. 2. Physical component *view*

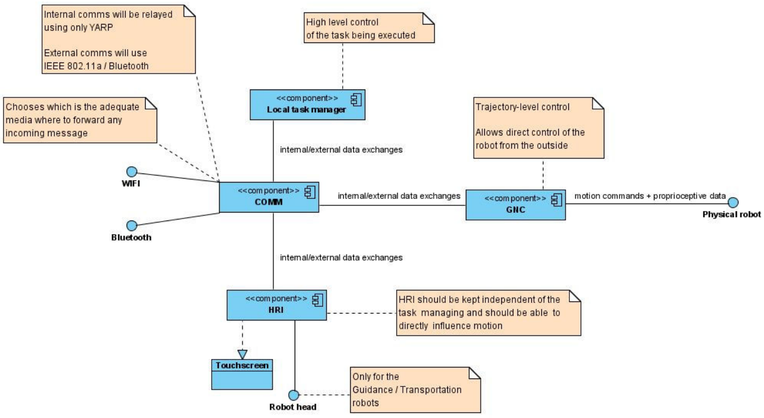

Fig. 3. Robot software component *view*

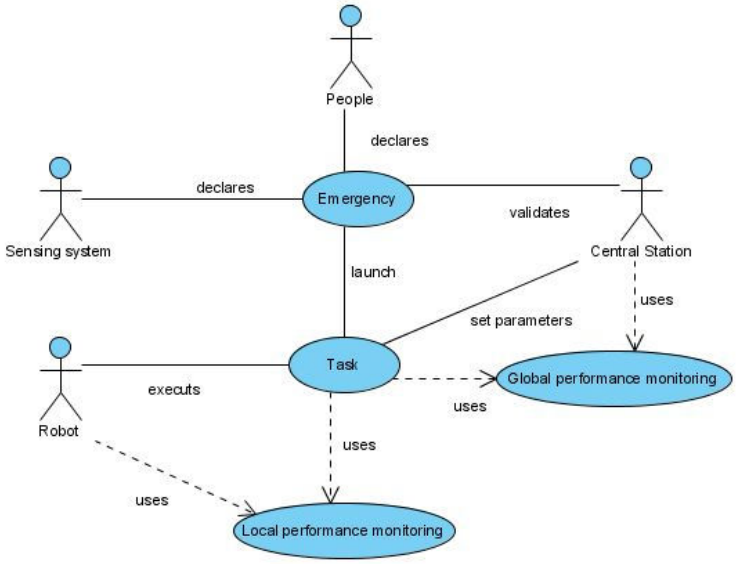

Fig. 4. Basis interaction control - Adjustment loops

Fig. 5. Artistic impression on natural human-robot interaction

experiment. After an emergency is declared by a human or by the sensing system, and validated by the CS a task is assigned to each robot and, if necessary, permanently adjusted by the robot and the CS.

B. The human view

The human *view* addresses the components related to the interactions between humans and the rest of the URUS system. In both of the experiments being developed within the project, the urban surveillance and the guidance/transportation of people and goods in an urban area it is fundamental that robots interact with humans in a "natural" form. Figure 5 shows an artistic impression of humanoid robots interacting with humans in the URUS test site.

The goal of having a system able to interact naturally with humans induces specific architectural concerns, namely, (i) robustness to disturbances, such as misinterpretation of sensor data, (ii) measurement of the quality of the interactions, (iii) flexibility, to allow addition/substitution/remotion/adaptation of some components to improve emotional response, and (iv) preservation of the coherence between the information provided by different interfacing tools.

The robustness concern is related to the synthesis of feature extraction algorithms and depends on the specific implementation of the components. Measuring the quality of interactions has long been a research topic. For example, the usability of interfaces such as cellphone keyboards has been assessed using the so called Fitts law. The mean time to complete a mobile phone typing task can be expressed as a linear function of this difficulty index, [20]. The Gibson model has been used in graphical interface design to predict the possible actions by the user, [10]. A linear expressiveness measure for data models in information systems has been defined in [5]. In what concerns expressiveness, the use of performance metrics to close feedback loops is yet uncommon in robotics. The "uncanny valley" paradigm, [15], suggests that having a wheeled robot, with few/none anthropomorphic features and moving in such a way that it can trigger emotional responses from humans that correspond to some specific value of the metric is a difficult problem. A supervised learning approach to estimate the expressiveness of robot motion has been presented in [21].

The flexibility and the information coherence concerns can

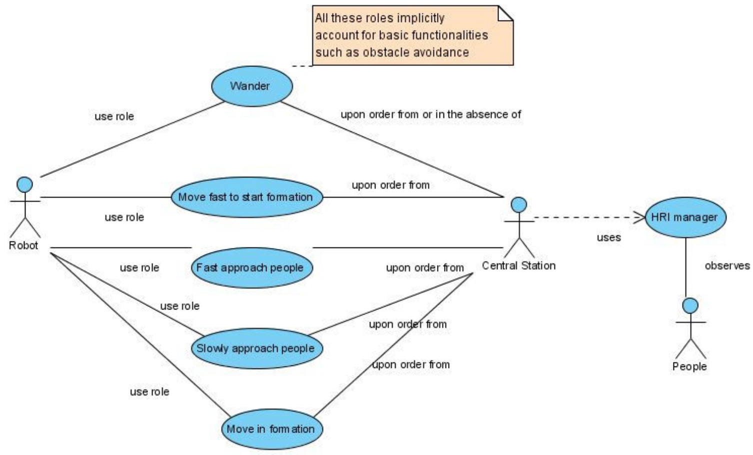

Fig. 6. Roles for the robots involved in the surveillance experiment

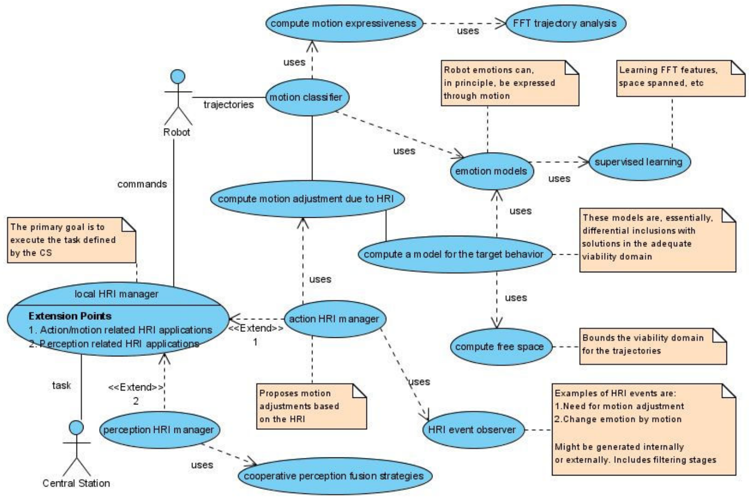

Fig. 7. Internal structure of a role

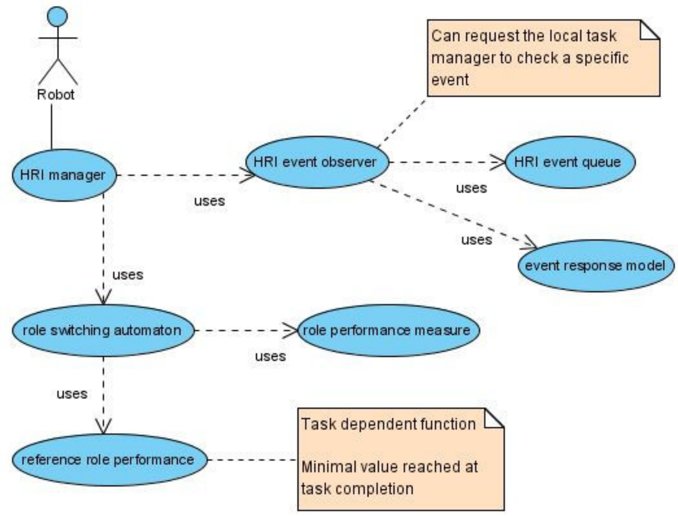

Fig. 8. Internal structure of the local HRI manager

be addressed in this *view*. The flexibility concern leads to the identification of a number of roles the actors directly related to HRI (humans, robots and CS) must play. Figure 6 shows a simplified use case diagram with a (non exhaustive) set of robot roles adequate for the surveillance experiment.

A typical situation that illustrates the importance of the information coherence concern arises when someone is waving to the imaging system to signal an emergency condition and, at the same time, uses a GUI, in a cellphone, to declare the same emergency[2] then it is necessary to ensure that both requests receive the adequate processing and reply, independently of their synchronism (or lack of it). Both the robots and the CS contain systems to preserve the coherence of the HRI information received that include components to queue and filter the arriving events related to HRI.

The CS has the additional tasks of maintaining at all times the coherence of behaviors by all actors, that is, given a sequence of generic events, namely related to HRI, the interchange of roles must be minimized. In small/medium scale systems keeping a centralized strategy to monitor all actors tends to be effective. In the URUS system any interaction related to HRI is monitored by the CS, even those occurring directly between humans and robots, e.g., via the bluetooth interface. Depending on the conditions exhibited by the surveillance scenario, the CS can parameterize the tasks assigned to each robot for instance to yield different global behaviors by the team.

A robot role contains a control structure to ensure that the robot moves accordingly while executing a mission. Figure 7 details a possible use case view for the internals of a generic role, related to the motion of a robot. The motion of the robot is decomposed in its Fourier components to assess its expressiveness. Fourier techniques can also be used to generate expressive motion (see for instance [26]). The local HRI manager receives requests from the local task manager asking for motion adjustments conform to the role. In the strict interpretation of the diagram, the HRI manager extends the basic capabilities of the local task manager.

A role is composed by specific motion, sensing, and interaction, strategies. A robot equipped with multiple roles amounts to a variable structure dynamic system and hence the local task manager, in charge of switching among roles, must account for situations of repeated fast switching. Figure 8 shows the internal structure of the local HRI manager.

This structure contains two main subsystems in charge of (i) processing any HRI related events arriving at the robot, and (ii) choosing the adequate role at each instance. It is inspired on standard results on the stability of variable structure systems, namely, the generalized version of the Lyapunov stability theorem (see for instance [22]) which roughly states that, under mild conditions, the switching between different models yields a stable system provided that the performance measure

[2]Generating this redundancy is an instinctive behavior commonly found in humans.

is kept bounded by a strictly decreasing continuous function.

The switching among roles is controlled by a finite state automaton, where each state is associated with one (or more) specific role. The notion of stability of an equilibrium state of a dynamic system can be identified with that of task completion, that is, the automaton must contain a state or a group of states whereto the decision making strategy must converge. Each task assigned to a robot defines a reference performance index that expresses at each instant the degree of completion of the task. The strategy to switch between roles compares this reference index to the indexes of the roles to assess potential stability problems while trying to converge to the equilibrium state.

IV. CONCLUSIONS

The paper presented a brief architectural description applicable to a generic networked robot system. The description was motivated by an urban surveillance application under development within the URUS project.

The architecture is of course inspired in the multiple works that have been described in the literature. Still, a number of innovative concepts have been used in this proposal, namely, the use of the IEEE 1471 methodology, the identification of the two views described with a model of human personality, and the use of concepts typical of dynamical systems analysis to define the internal structure of some of the systems.

The architecture discussed develops in two views. The first view corresponds to a functional layer that ensures that a task assigned to the system can be carried out successfully even if human interaction tools are poor. The second view sets an additional layer on top of the functional layer, that is exclusively concerned with human-robot interaction. Structural elements and key concerns have been identified for this layer.

ACKNOWLEDGMENTS

This work was supported by the European Project FP6-2005-IST-6-045062-URUS, and ISR/IST plurianual funding through the POS_Conhecimento Program that includes FEDER funds.

This work profits from discussions with the whole URUS team.

REFERENCES

[1] http://wiki.robot-standards.org/index.php/Main_page. Accessed June 2008.

[2] http://www.oasis-open.org/specs/index.php. Accessed June 2008.

[3] http://www.okiproject.org/. Accessed November 2007.

[4] Alexander M. Meystel and James S. Albus. *Intelligent Systems: Architecture, Design, and Control*. Wiley Interscience, 2002.

[5] M. Ceruti and S. Robin. Infodynamics: Analogical analysis of states of matter and information. *Int. Journal on Information Sciences*, (177):969987, 2007.

[6] EURON Network of Excellence on Information Society Technologies Priority. DR.1.1, Research Roadmap V1. Technical report, EURON, 2005. University of Karlsruhe.

[7] Evan Woo, Bruce A. MacDonald, and Félix Trépanier. Distributed Mobile Robot Application Infrastructure. In *Procs of the Int. Conf. on Intelligent Robots and Systems (IROS 2003)*, pages 1475–1480, October 2003. Las Vegas.

[8] Hrand Aghazarian, Paolo Pirjanian, Paul Schenker, and Terrance Huntsberger. An Architecture for Controlling Multiple Robots. www.nasatech.com/Briefs/Oct04/NPO30345.html, webpaper accessed June 2006.

[9] IEEE. IEEE Recommended Practice for Architecture Description of Software-Intensive Systems, 2000. E-ISBN 0-7381-2519-9, ISBN 0-7381-2518-0.

[10] John M. Carroll. Toward a Multidisciplinary Science of Human-Computer Interaction. In *HCI Models, Theories and Frameworks*. Morgan-Kaufmann, 2003.

[11] Julio Rosenblatt. DAMN: A Distributed Architecture for Mobile Navigation. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2/3):339–360, 1997.

[12] Kurt Konolidge. DARPA Software for Distributed Robotics, Centibots Large Scale Robot Teams. Technical Report 2002-12-01, 2002.

[13] A. Maslow. A Theory of Human Motivation. *Psychological Review*, 50(4):370–396, 1943.

[14] Matt Long, Aaron Gage, Robin Murphy, and Kimon Valavanis. Application of the Distributed Field Robot Architecture to a Simulated Demining Task. In *Procs of the 2005 IEEE Int. Conf. on Robotics and Automation (ICRA 05)*, 2005. Barcelona, Spain.

[15] M. Mori. Bukimi no tani the uncanny valley. *Energy*, 7(4):33–35, 1970. Translation by Karl F. MacDorman and Takashi Minato, from the Japanese version, in MacDorman, Karl F. (2005). Androids as an experimental apparatus: Why is there an uncanny valley and can we exploit it? CogSci-2005 Workshop: Toward Social Mechanisms of Android Science, 106-118.

[16] Oliver Schorr, Nobuhiko Hata, Andrew Bzostek, Rajesh Kumar, Catherina Burghart, Russell H. Taylor, and Ron Kikinis. Distributed Modular Computer-Integrated Surgical Robotic Systems: Architecture for Intelligent Object Distribution. In *Procs of the 3rd Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, page 979987, 2000.

[17] R.T.Laird, H.R. Everett, G.A. Gilbreath, T.A. Heath-Pastore, and R.S. Inderieden. MDARS: Multiple Robot Host Architecture. In *Procs. of the 22nd Annual Technical Symposium and Exhibition (AUVS '95)*, July 1995. In Association of Unmanned Vehicle Systems, Washington DC.

[18] A. Sanfeliu and J. Andrade-Cetto. Ubiquitous networking robotics in urban settings. In *Procs. of the IEEE/RSJ IROS Workshop on Network Robot Systems*, pages 14–18, October 2006. Beijing, China.

[19] Satoru Satake, Hideyuki Kawashima, and Michita Imai. LACOS: Context Management System For a Sensor-Robot Network. In *Procs. of the 10th IASTED International Conference on Robotics and Applications (RA 2004)*, pages 160–165, 2004.

[20] Scott Mackenzie. Motor Behavior Models for Human-Computer Interaction. In *HCI Models, Theories and Frameworks*. Morgan-Kaufmann, 2003.

[21] J. Sequeira. Measuring the expressiveness of wheeled mobile robots. In *Procs. of the EURON European Robotics Symposium (EUROS 2008)*, 2008. Prague, Czech Republic, March 26-27.

[22] G. Smirnov. *Introduction to the Theory of Differential Inclusions*, volume 41 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

[23] Stefan Enderle, Stefan Sablatnoeg, and Hans Utz. MIRO Middleware for Robots. http://smart.informatik.uni-ulm.de/MIRO/, website accessed June 2006.

[24] Tetsuya Taira and Nobuyuki Yamasaki. Functionally Distributed Control Architecture for Robot Systems. *Journal of Robotics and Mechatronics*, 16(2):217–224, 2004.

[25] TH. Laengle and T.C. Lueth. Decentralized Control of Distributed Intelligent Robots and Subsystems. In *Procs of the IFAC Symposium on Artificial Intelligence In Real Time Control*, October 1994. Valencia, Spain.

[26] Unuma, M. and Anjyo, K. and Takeuchi, R. Fourier Principles for Emotion-Based Human Figure Animation. In *Procs. of the ACM*, pages 91–96, 1995.

[27] Yuan-hsin Kuo and Bruce A. MacDonald. Designing a Distributed Real-time Software Framework for Robotics. In *Procs. of the Australasian Conf. on Robotics and Automation*, December 2004. Canberra, Australia.