PEIS Ecologies: Ambient Intelligence meets Autonomous Robotics

Alessandro Saffiotti & Mathias Broxvall

AASS Mobile Robotics Laboratory, University of Örebro, Sweden
{asaffio,mbl}@aass.oru.se
www.aass.oru.se/˜peis

Abstract

A common vision in the field of autonomous robotics is to create a skilled robot companion that is able to live in our homes and perform physical tasks to help us in our everyday life. Another vision, coming from the field of ambient intelligence, is to create a network of intelligent home devices that provide us with information, communication, and entertainment. We propose to combine these two visions into the new concept of an ecology of networked Physically Embedded Intelligent Systems (PEIS). In this paper, we define this concept, and illustrate it by describing an experimental system that involves real robotic devices.

1 Introduction

> *Johanna is 76 years old, and she lives in a small house. Soon before she wakes up, her fridge realizes that there is little milk left and sends a request for a new bottle to the local store. Johanna's autonomous trolley goes to the store and fetches the milk together with the usual newspaper. When Johanna gets out of bed, her motion is detected and the coffee machine starts to brew coffee. A team of kitchen robots bring bread, butter, jam and coffee on the table. When Johanna walks out of the bathroom, she finds her breakfast ready. She hears the cleaning robot that cleans the bathroom after her.*

In a decade or two, **physically embedded intelligent systems** (PEIS) will become part of our daily experience and improve the quality of life for every citizen, and especially for those in need of special physical and/or cognitive care like senior citizens. Examples of such systems include autonomous cleaners, intelligent appliances, smart user interfaces, and dedicated devices (robots) able to carry objects or perform assistive tasks.

In the vision that inspires our work, PEIS do not exist and operate in isolation: they form an **ecology** of communicating and cooperating PEIS, both inside and outside our houses. The overall functionality of the full system emerges from the interaction between a number of simpler units. The functionality of each individual unit is improved by this interaction. As an example, the table setting robots in the vignette above do not need to use their sensors to detect the position, shape, and weight of the milk bottle in the fridge in order to grasp it — a task which proved to be surprisingly difficult in years of robotic research [9]. Instead, the fridge, or even the bottle itself enriched with a micro-PEIS, would hold this information and communicate it to the robot.

The idea of networking everyday objects and appliances in the environment has been actively pursued in the field of Ambient Intelligence [5]. However, emphasis has usually been on the creation and delivery of information, and not on the performance of physical tasks.

The concept of PEIS-Ecology puts together insights from the fields of ambient intelligence and autonomous robotics to generate a radically new approach to building assistive, personal, and service robots. Most current approaches to building a "robot companion" aim at building one isolated robotic device (often humanlike) empowered with extraordinary abilities for perception, action, and cognition (e.g., [14]). By contrast, the PEIS-Ecology approach redefines the very notion of a *robot* to encompass the entire environment. Perception and manipulation of objects are thus replaced by direct communication between sub-systems in the environment. In the PEIS-Ecology vision, the robot would disappear in the environment quite in the same way as computers should disappear according to the well known vision of ubiquitous computing [15].

2 Converging Technologies

The development of a PEIS-Ecology relies on the integration between three main fields: robotics, artificial intelligence, and ubiquitous computing.

First, individual PEIS in the ecology are based on *robotic technology*, taken out from the current robotic systems and used inside specialized devices. Many of these PEIS's would not look like today's ordinary robots more than a cell phone looks like a yesterday's mainframe computer.

Second, PEIS need a layer of intelligence, or cognition, in order to deal with missing or uncertain information, to modify their behavior according to the current tasks and situation, and to detect and possibly recover

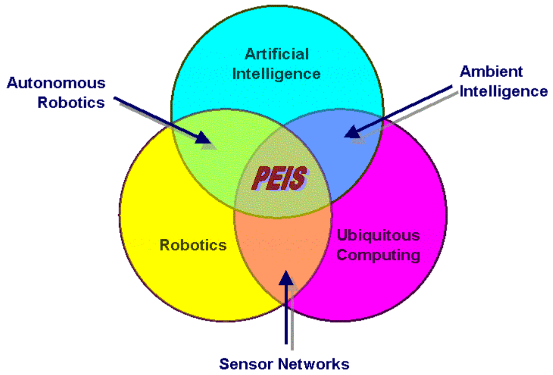


Figure 1: The study of PEIS-Ecologies lays at the intersection of several different fields.

from exceptional situations and failures. The introduction of cognition inside a PEIS is based on *artificial intelligence* (AI) technology. Cognition should not only be present inside the individual PEIS, but would also emerge in the whole ecology as a result of the cooperation between them.

Finally, all the individual PEIS need to be connected into a whole networked system. It is through these connections, and through the establishments of mechanisms for communication and cooperation, that a set of individual PEIS becomes a PEIS-Ecology. The study of the mechanisms to connect multiple, heterogeneous, embedded computing devices will profit from the current developments in the field of *ubiquitous computing*.

While the fields of robotics, AI, and ubiquitous computing have produced impressive results in isolation, the crucial problem of the *integration* between these fields has often been an orphan in this development. PEIS-Ecologies can be seen as a new research area at the intersection between these fields, as shown in Fig. 1.

The above integration problem is far from being trivial due to fundamental differences between these fields. For instance, techniques developed in AI are usually based on symbol-level representations, and they take a global view (both in space and time) on a discretized world. Techniques developed in robotics are usually based on signal-level representations, and they take a local view (both in space and time) on a continuous world. And techniques developed in ubiquitous computing usually do not take into account the aspects of physical embedding (perception and action) and of intelligence. Although some effort has recently been done in the pairwise integration of these three fields (e.g., autonomous robotics, ambient intelligence, sensor networks), very little research exists that integrates all of the three fields. To the best of our knowledge, the only two efforts in this direction are the Japanese Network Robot project [10], and the project presented here.

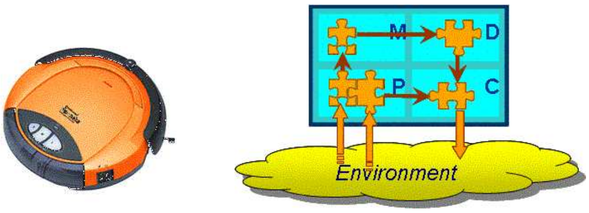


Figure 2: A commercial autonomous vacuum cleaner (left), and its view as a PEIS (right).

While the former project has a markedly application-oriented flavor, our work puts a greater emphasis on the conceptual integration between the above areas.

3 The Concept of PEIS-Ecology

The central notion of our integration effort is the concept of an *Ecology of Physically Embedded Intelligent System*, or PEIS-Ecology. The main distinctive features of this concept are as follows.

1. Any robot in the environment is abstracted by the *uniform notion of a* PEIS, a physical device which includes a number of functional components. The term “robot” is taken here in its most general interpretation: a computerized system interacting with the environment through sensors and/or actuators. A PEIS can be as simple as a toaster and as complex as a humanoid robot.

2. All PEIS can be connected by a *uniform cooperation model*, based on the notion of linking functional components. More precisely, each participating PEIS can use functionalities from other PEIS in the ecology in order to compensate or to complement its own. The power of the PEIS-Ecology does not come from the individual power of its constituent PEIS, but it emerges from their ability to interact and cooperate.

As an illustration of this concept, consider the autonomous vacuum cleaning robot shown in Figure 2. This robot can be seen as a PEIS which incorporates a number of functional components for perception (P), modeling (M), deliberation (D) and control (C). By itself, this robot cannot devise and execute complex cleaning strategies because it cannot reliably know its own position in the home. In fact, the simple perceptual components in the robot may be enough to perform local navigation and obstacle avoidance, but not to perform reliable global localization.

Suppose, however, that the home is equipped with an ambient monitoring system that uses input from a set of static cameras to reconstruct the state of the environment, including the position of the vacuum cleaner.

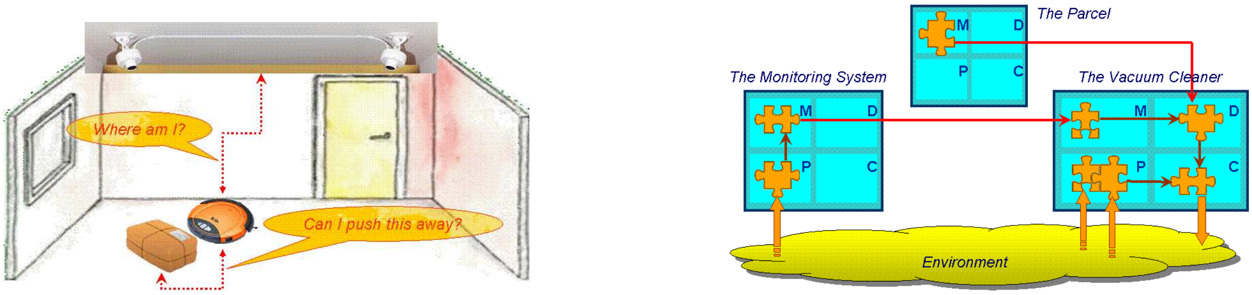


Figure 3: Left: a simple cooperation scenario. Right: the same scenario viewed as a PEIS-Ecology.

This monitoring system can be seen as a PEIS. Then, we can combine the vacuum cleaner and the monitoring system into a simple PEIS-Ecology, in which the former provides the latter with a global localization functionality. The vacuum cleaner can use this functionality to decide a smarter navigation and cleaning strategy.

Suppose then that the vacuum cleaning robot encounters an unexpected parcel laying on the floor. The robot needs to know its properties (e.g., weight and fragility) in order to decide weather it can push it or not. This information may be difficult or impossible to acquire using the robot's sensor. However, if the parcel itself contains an IC tag holding information about its content, then the parcel can be seen as a PEIS and it can join the ecology, thus delivering this information directly to the robot. The resulting ecology is shown in Fig. 3.

In slightly more precise terms, we define a PEIS to be a set of inter-connected software components residing in one physical entity. Each component may include links to sensors and actuators that connect it to the physical environment, as well as input and output ports that connect it to other components in the same PEIS or in other PEIS. A PEIS may incorporate components for perception, modeling, deliberation and control functionalities, but simple PEIS may include only a few of these functionalities.

We then define a PEIS-*Ecology* to be a collection of inter-connected PEIS, all embedded in the same physical environment. We assume that there is a (possibly heterogeneous) underlying communication mechanism to provide the needed connections.

Finally, we call *configuration* of a PEIS-Ecology the set of connections between components within and across the PEIS in the ecology. Note that the same ecology can be configured in many different ways depending on the current context. Relevant contextual aspects here include the current goals, situation, and resources. The PEIS-Ecology can be seen as a general purpose robotic servant distributed in the home, whose shape and functions change dynamically according to the context. In the above scenario, if the information from the monitoring system becomes unavailable, e.g. because of darkness, then the vacuum cleaner should revert to use the location information provided by its own odometric system. An approach that uses AI planning techniques to automatically generate such configurations for a given context is presented in [8].

It is interesting to note that a PEIS-Ecology can conceivably be seen as an *ecology* according to the usual definition of this term:

> *The study of the relationships between living organisms and their environment.*

The main points in this definition are: the central role played by both the organisms and the environment; and the emphasis on the study of their relationships. In a similar way, a PEIS-Ecology is characterized both by the relationship among PEIS entities (through communication and cooperation), and by the relationship between these entities and their environment (through perception and action).

4 An Experimental Test-Bed

In order to verify the viability of the PEIS-Ecology framework, we have developed an experimental test-be in which PEIS-Ecologies can be implemented and tested on several scenarios. The test-bed consists of the following ingredients:

- The PEIS-Home, an home-like environment that incorporates a number of PEIS;
- The PEIS reference architecture, in which PEIS-Ecologies can be defined, implemented and run;
- A set of PEIS-Components, including perception, reasoning, and action components.

In this section, we detail the above ingredients, and show a simple example of a real PEIS-Ecology similar to the one shown in Fig. 3 above.

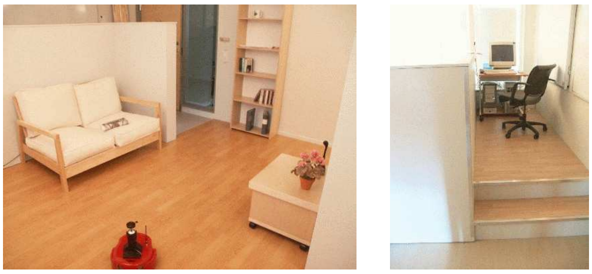


Figure 4: Two views of the PEIS-Home.

4.1 The PEIS-Home

The PEIS-Home is an experimental environment that looks like a typical Swedish mini-apartment of about $25\,\mathrm{m}^2$. It consists of a living room, a bedroom, and a small kitchen. The walls are only $1.40\,\mathrm{m}$ high, so that the observers to get an easier view of the inside of the PEIS-Home. The ceiling has been lowered to a standard height of $2.40\,\mathrm{m}$: the space above the ceiling is used for cables, computing equipment, and WLAN access points. Four web-cameras have been mounted on the ceiling, connected to a localization system.

Beside the main apartment there is an "observation deck", a small elevated area from which the experimenters and the visitors can observe and control the behavior of the full PEIS-Ecology in the PEIS-Home. Fig. 4 shows a partial view of the apartment (left) and of the observation deck (right).

4.2 The PEIS-Architecture

The main technical ingredient needed to implement a PEIS-Ecology is a reference architecture that enables the communication and coordination between all the components in the ecology. Our reference architecture combines an event-based and a tuple-based communication models within a component framework.

The communication model. In choosing a communication model, we have to keep in mind three factors that characterize a PEIS-Ecology: (1) there may be a large number of PEIS exchanging data; (2) PEIS can join and leave the ecology at any moment, and they can be temporarily unavailable (e.g., out of radio range); and (3) some interactions can be tightly coupled. These factors suggest that a request/response communication model would not be adequate in our case. For example, a client that requests instantaneous updates of information would need to continuously poll the information providers, leading to network overload and congestion. Instead, we have opted for a model based on the integration of a distributed tuple-space with an event mechanism. Hybrid approaches combining tuple-spaces and events are increasingly used both in ambient intelligence systems [2, 13], in sensor networks [1, 4], and in autonomous robots [3].

To allow for more efficient and simple implementations of the distributed tuple space, all tuples are restricted to be of the form:

```
<peisID, compID, key, val0, ..., valN>
```

where `peisID` and `compID` are uniquely assigned to every PEIS and every component inside it, and `key` is the tuple key. This convention simplifies the resolution of conflicts for multiple producers of the same key.

The tuple-space is endowed with the usual `insert` and `read` operations [16]. In addition, we define event-based primitives `subscribe` and `unsubscribe`, by which a PEIS can signal its interest in a given key. When an insert operation is performed, all subscribers are notified. Management of subscription and notifications is done by the middle-ware, in a way which is transparent to the PEIS.

The PEIS-ontology. In order to share information, the various components in the PEIS-Ecology need to know which tuples exist, what data they contain, and the semantics of those data. Said differently, there is a need for a *shared ontology* in the PEIS-Ecology.

The ontology is stored in the tuple-space, in tuples associated to the reserved key `Description`. Whenever a PEIS joins the ecology, it publishes an XML description of its components, including their input ports, output ports, dependencies, type of data, etc. These can be used, for instance, by a deliberation component to automatically generate possible configurations to perform a given task [8]. From the point of view of the tuple-space, descriptions are manipulated just like any other tuple. In particular, PEIS-components can be notified of any changes in the composition of the ecology by subscribing to the `Description` key.

The PEIS-kernel. The above mechanisms are implemented in the PEIS-*kernel*. This is a software layer that resides on the top of the operating system, and provides a common abstraction by which each PEIS-component can read and write data to the distributed tuple space. The PEIS-kernel provides platform-independent and network-independent communication primitives, stores a local portion of the distributed tuple-space, and manages the events. The PEIS-kernel must be present on every CPU on which some PEIS-component is run.

Fig. 5 shows one possible way to implement the PEIS-Ecology in Fig. 3 above. (We omit the parcel for simplicity.) In this example, the two PEIS-components in the monitoring system are run as two processes on one CPU; while the five PEIS-components in the vacuum cleaner are run as three processes which are distributed over two CPU's inside the cleaner. Communication between PEIS-components that run as separate

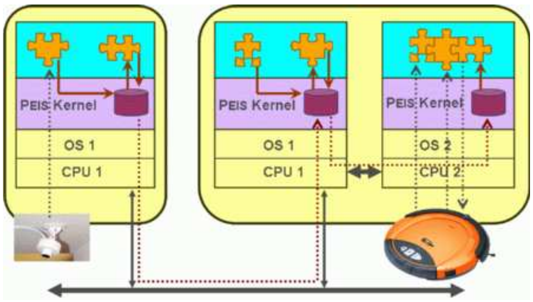


Figure 5: An implementation of the above ecology.

processes always go through the PEIS-kernel, and are in the form of exchange of tuples. The link between tuples stored inside different PEIS-kernels is realized by the PEIS-kernels themselves in a way which is transparent to the PEIS-components.

Our current implementation of the PEIS-kernel takes the form of a Linux run-time library, which performs network discovery, creates and maintains an ad-hoc P2P network over TCP/IP, and takes care of providing a shared tuple-space between the different PEIS.

4.3 The PEIS-Components

We have implemented a number of robotic software components that utilize the PEIS-framework to provide and share functionalities. The main components are:

The Thinking Cap component. This is an architecture for autonomous robot control based on fuzzy logic [12]. It provides a mobile robot with advanced navigation functionalities. It responds to tuples with key `ExecGoal`, which provide a navigation goal. It produces tuples with keys `Odometry`, `ExecStatus`, etc, which reflect the robot status.

The PTLplan component. This is a conditional planner used for high-level task planning [6]. PTLplan can reason about uncertainty and about observation actions. It responds to goal tuples with key `Task` and information tuples with keys `WorldStatus` and `ExecStatus`. It produces tuples with keys `ExecGoal` to indicate new high-level actions to be performed (typically by the ThinkingCap).

The Player component. This is a PEIS-wrapped instance of the player program [11], which provides a low-level interface between the robot's sensors and actuators and the PEIS-Ecology's tuple-space. It reads robot command tuples (e.g., `VelCommand` and `TalkCommand`), and generates sensor reading tuples (e.g., `Odometry` and `SonarRange`).

The localization component. This is a simple local positioning system based on [7]. It uses the web-cameras mounted on the ceiling to track a colored object using triangulation, and produces `AbsolutePos` tuples. In our experiments, we have equipped the tracked robot with a colored "hat".

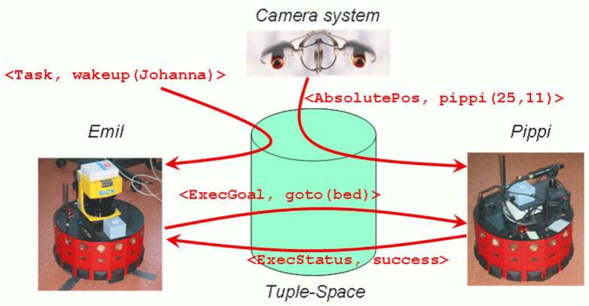


Figure 6: Tuples exchanged in our experiment. The tuple-space is physically distributed across all PEIS's.

The PEIS-Ecology monitor. This can be used by the experimenters to dynamically observe, and possibly influence, the internal state of the PEIS-Ecology. The monitor includes a 3D visualization tool and a record/playback tool.

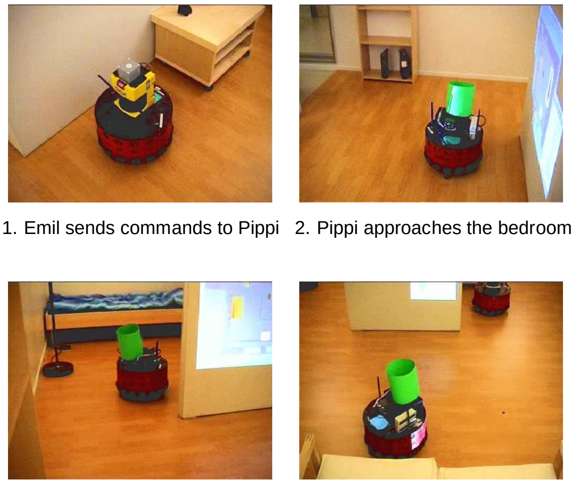
4.4 A concrete example

To better illustrate the working of a PEIS-Ecology, we show a simple experiment in which a robot is given the task to go and wake up Johanna. The PEIS-Ecology implemented for this experiment consists of the four PEIS:

- The Pippi PEIS, a Magellan Pro robot running an instance of the ThinkingCap component and of the Player component.
- The Emil PEIS, a Magellan Pro robot running an instance of the ThinkingCap, the Player, and the PTLplan components.
- The tracking PEIS, running an instance of the localization component.
- The monitor PEIS, running an instance of the PEIS-monitor component.

The scenario of this experiment is as follows. Emil receives the high-level goal to wake up Johanna. She decides to delegate the execution of this task to Pippi,[1] and generates a plan for Pippi using PTLplan. The plan consists of three actions: GoTo(bed), Talk(wakeup), GoTo(sofa). Emil sends the first action to Pippi, who uses ThinkingCap and Player to execute it. In order to better keep track of her own position in the environment, Pippi uses the information provided by the tracking PEIS. If Pippi occasionally exits the field of

[1] The reasons why Emil receives this goal, and why she decides to delegate execution to Pippi, are outside the scope of this experiment.



1. Emil sends commands to Pippi 2. Pippi approaches the bedroom

3. Pippi exits the bedroom 4. Task completed

Figure 7: Four snapshots taken during an actual run.

view of the cameras, she reverts to using the (unreliable) self-location information from its internal odometry. When Pippi reaches the bed, she notifies this event to Emil, who then sends her the next action, and so on. When Pippi finally reaches her home position, she notifies Emil and the task is completed.

All the above communications and synchronizations happen by the exchange of the relevant tuples through the distributed tuple-space, and are mediated by the PEIS-kernel. This scenario is graphically illustrated in Fig. 6. (The PEIS-monitoring system is not included for simplicity.) Fig. 7 shows four snapshots during an actual execution of this scenario.

5 Conclusions

PEIS-Ecologies offer a new paradigm to develop pervasive robotic applications that will bring robotic technologies inside our homes and improve the quality of our life. PEIS-Ecologies combine ideas, theories and techniques from the fields of ambient intelligence and autonomous robotics, and bring a new dimensions to both fields. While the development of PEIS-Ecologies reported in this paper is still at an early stage, we believe that this new paradigm will be extremely rich of both theoretical and practical consequences.

Among our many current directions of development, we mention: the extension of the current experimental system to incorporate a wider variety of PEIS; the inclusion of small embedded devices (e.g., smart objects and RFID-tagged objects) in our PEIS-Ecologies; the automatic configuration and re-configuration of a PEIS-Ecology to account for changes in the ecology, in the environment, or in the system's goals; the study of human-robot interaction in the PEIS-Ecology framework; and the study of robot-object interaction using a combination of traditional means (sensors and actuators) and PEIS-based communication.

Acknowledgments

This work has been supported partly by the Swedish KK Foundation, and partly by ETRI (Electronics and Telecommunications Research Institute, Korea) through the project "Embedded Component Technology and Standardization for URC (2004-2008)". Many thanks to Donatella Guarino, Marco Gritti and Beom-Su Seo for many enlightening discussions.

References

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. In *Proc. of the 17th ACM Symp. on Operating systems principles*, pages 186–201, 1999.

[2] D. Arregui, C. Fernstrom, F. Pacull, and J. Gilbert. Stitch: Middleware for ubiquitous applications. In *Proc. of the Smart Object Conf.*, Grenoble, France, 2003.

[3] D.A. Cáceres, H. Martínez, M.A. Zamora, and L.M. Tomás. A real-time framework for robotics software. In *Int Conf on Computer Integrated Manufacturing*, 2003.

[4] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proc. of the 18th ACM Symp. on Operating systems principles*, pages 146–159, 2001.

[5] IST Advisory Group. Ambient intelligence: from vision to reality. In G. Riva, F. Vatalaro, F. Davide, and M. Alca niz, editors, *Ambient Intelligence*. IOS Press, 2005.

[6] L. Karlsson. Conditional progressive planning under uncertainty. In *Proc. of the 17th IJCAI Conf.*, 2001.

[7] A. Lilienthal and T. Duckett. An absolute positioning system for 100 euros. In *Proc. IEEE Int. Workshop on Robotic Sensing*, Örebro, Sweden, 2003.

[8] R. Lundh, L. Karlsson, and A. Saffiotti. Can Emil help Pippi? In *Proc. of the ICRA-05 Workshop on Cooperative Robotics*, Barcelona, ES, 2005. Online at www.aass.oru.se/˜asaffio/.

[9] R. Molfino, editor. *Proc. of the 1st Int. Conf. on Intelligent Manipulation and Grasping*, Genova, Italy, 2004.

[10] Network Robot Forum. www.scat.or.jp/nrf/English/.

[11] Player/Stage Project. playerstage.sourceforge.net/.

[12] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.

[13] F. Siegemund. A context-aware communication platform for smart objects. In *Proc. of the Int. Conf. on Pervasive Computing*, 2004.

[14] The Cogniron Consortiom. The cognitive robot companion. www.cogniron.org.

[15] M. Weiser. The computer for the 21st century. *Scientific American*, 9:66–75, 1991.

[16] P. Wyckoff, S.W. McLaughry, T.J. Lehman, and D.A. Ford. T-spaces. *IBM Systems Journal*, 37(3), 1988.