# The PEIS-Ecology Project: Vision and Results

A. Saffiotti, M. Broxvall, M. Gritti,
K. LeBlanc, R. Lundh, J. Rashid
AASS Mobile Robotics Lab
Örebro University, S-70182 Örebro, Sweden
{asaffio,mbl}@aass.oru.se

B.S. Seo, Y.J. Cho
Electronics and Telecommunications Research Institute
161, Gajeong-dong, Yuseong-gu, Daejeon, Korea
{bsseo,youngjo}@etri.re.kr

*Abstract*— **The vision of an Ecology of Physically Embedded Intelligent Systems, or PEIS-Ecology, combines insights from the fields of autonomous robotics and ambient intelligence to provide a new approach to building robotic systems in the service of people. In this paper, we present this vision, and we report the results of a four-year collaborative research project between Sweden and Korea aimed at the concrete realization of this vision. We focus in particular on three results: a robotic middleware able to cope with highly heterogeneous systems; a technique for autonomous self-configuration and re-configuration; and a study of the problem of sharing information of both physical and digital nature.**

## I. INTRODUCTION

The PEIS-Ecology approach belongs to a trend which is becoming rather popular in the area of home and service robotics: to abandon the idea of having one extremely competent isolated robot acting in a passive environment, in favor of a network of cooperating robotic devices embedded in the environment [1], [2], [3], [4]. In the PEIS-Ecology approach, advanced robotic functionalities are not achieved by the development of extremely advanced robots, but through the cooperation of many simple robotic components.

The PEIS-Ecology approach takes an ecological view of the robot-environment relationship, in which the robot and the environment are seen as parts of the same system, engaged in a symbiotic relationship toward the achievement of a common goal, or equilibrium status. We assume that robotic devices (or PEIS, for Physically Embedded Intelligent Systems) are pervasively distributed throughout the working space in the form of sensors, actuators, smart appliances, RFID-tagged objects, or more traditional mobile robots; and that these PEIS can communicate and collaborate with each-other by providing information and by performing actions. Humans can also be included in this approach as another species of PEIS inside the same ecosystem.

The PEIS-Ecology approach was developed in the context of a collaborative research project between the University of Örebro, Sweden, and ETRI (Electronic and Telecommunication Research Institute), Korea, run from October 2004 to December 2007. In this paper, we outline the vision of PEIS-Ecology, and we present the main results achieved in our collaborative project with respect to three of the major scientific and technological challenges entailed by its realization: (1) dealing with the inherent heterogeneity of a PEIS-Ecology; (2) allowing a PEIS-Ecology to self-configure to perform a given task in a given context, and to automatically re-configure if the context changes; and (3) sharing information of different types among PEIS, while crossing the border between physical and digital world.

This paper is intended to provide, in one single place, a broad overview of the project vision and results. Because of this, we shall not dive into technical details here. Technical papers, source code, and demonstration videos can be found at the project web site [5].

## II. THE PEIS-ECOLOGY VISION

The vision of PEIS-Ecology, originally introduced by Saffiotti and Broxvall [6], combines insights from the fields of autonomous robotics and ambient intelligence to provide a new solution to building intelligent robotic systems in the service of people.

### A. The concept

Our vision builds upon the following ingredients.

First, any robot in the environment is abstracted by the *uniform notion* of PEIS[1] (Physically Embedded Intelligent System). The term "robot" is taken here in its most general interpretation: any device incorporating some computational and communication resources, and able to interact with the environment via sensors and/or actuators. A PEIS can be as simple as a toaster or as complex as a humanoid robot. In general, we define a PEIS to be a set of inter-connected software *components* residing in one physical entity. Each component can be connected to sensors and actuators in that physical entity, as well as to other components in the same PEIS or in other PEIS.

Second, all PEIS are connected by a *uniform communication model*, which allows the exchange of information among the individual PEIS-components, while hiding the heterogeneity of the PEIS and of the physical communication layers. In practice, we use a distributed communication model that combines a tuple-space with an event mechanism.

Third, all PEIS in an ecology can cooperate by a *uniform cooperation model*, based on the notion of linking functional

Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-08). Nice, France, September 22–26, 2008.

---

[1] PEIS is pronounced /peɪs/ like in 'pace'.

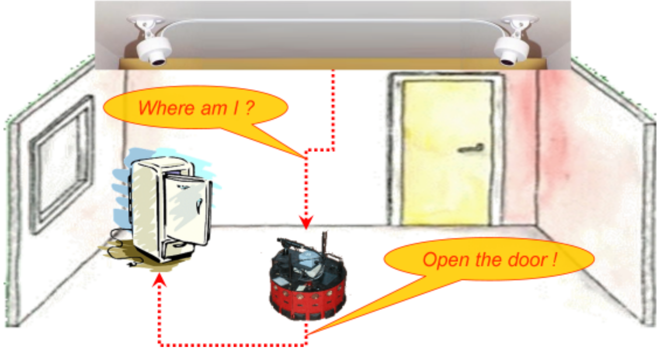


Fig. 1. A simple example of PEIS-Ecology.

components: each participating PEIS can use functionalities from other PEIS in the ecology to complement its own. Functionalities here are meant to be modules that produce and consume information, and may interact with the physical environment by means of sensors and actuators. Typically, but not necessarily, functionalities are in one-to-one correspondence to the software components in a PEIS.

Finally, we define a PEIS-Ecology to be a collection of inter-connected PEIS, all embedded in the same physical environment.

As an illustration of these concepts, consider a home robot with the task of grasping a milk bottle from the fridge — see Figure 1. In a classical approach, the robot would use its sensors to acquire information from the environment, e.g., to self-localize, and to acquire the relevant parameters of the fridge handle and of the milk bottle. It would then use its actuators to manipulate the environment, e.g., to open the fridge door and to grasp the milk bottle. In a PEIS-Ecology, by contrast, the robot would ask (some of) the needed information from the environment, e.g., it would get its position from cameras in the ceiling; and it would get the shape, weight, and grasping points of the milk bottle from the bottle itself, equipped with a mote or an RFID tag. It would also ask the environment to perform (some of) the needed actions, e.g., it would ask the fridge to open its door.

Given a PEIS-Ecology, we call *configuration* a set of components in the ecology together with a set of connections between them. Intuitively, a configuration is a way to connect (some of) the components in the ecology in order to perform a given task. Importantly, the same ecology can usually be configured in many different ways, depending on the context — e.g., depending on the current task, the environmental situation, and available resources. In the above example, if the robot exits the field of view of the ceiling cameras, then the ecology may be reconfigured to let the robot use a different PEIS for localization, e.g., a camera on the fridge or the robot's own odometric system.

### B. The implications

A PEIS-Ecology redefines the very notion of a *robot* to encompass the entire environment: a PEIS-Ecology may be seen as a "cognitive robotic environment" in which perception, actuation, memory, and processing are pervasively distributed in the environment. The complex functionalities of this environment are not determined in a centralized way, but they emerge from the co-operation of many simpler, specialized, ubiquitous PEIS devices. The number and capabilities of these devices do not need to be known *a priori*: new PEIS can join or leave the ecology at any moment, and their existence and capabilities should be automatically detected by the other PEIS.

Similarly to a natural ecosystems, a PEIS-Ecology is characterized by: (i) the relationship between living entities (PEIS) and the environment (through sensors and actuators), (ii) the relationship among these entities (through the cooperation model), and (iii) the notion that complex behavior emerges from the interaction of simpler units. Moreover, a PEIS-Ecology is intended to include different *species* of entities in symbiotic interaction. In the PEIS-Ecology vision, humans can constitute one of the species who participate in the ecology, and interact with the other PEIS in it.

The PEIS-Ecology approach simplifies many of the difficult problems of current autonomous robotics by replacing complex on-board functionalities with simple off-board functionalities plus communication. In the milk example above, the global localization of the robot is easily achieved by the static cameras; and the best way to access the properties of the milk box is to store those properties in the box itself. The PEIS-Ecology approach can also help us to address problems which are beyond the capabilities of current robotic systems. An example of this type is reported in [7], where a PEIS-Ecology is used to solve a home monitoring task involving the use of olfaction, which would be hard to solve otherwise due the current limitations of mobile olfaction.

The PEIS-Ecology approach can also bring a number of pragmatic benefits. A PEIS-Ecology is intrinsically modular, flexible and customizable. Users would only need to acquire new robotic components as needed, e.g., starting with just a simple robotic vacuum cleaner and adding new PEIS devices according to their changing needs and desires. Thus, the PEIS-Ecology approach is likely to provide an affordable and acceptable road to include robotic technologies in everyday environments. Since each new PEIS can combine its functionalities with those of the already exiting ones, the value of the whole PEIS-Ecology is more than the sum of the values of the individual PEIS in it.

### C. Realizing the vision

As noted in the Introduction, the concrete realization of the PEIS-Ecology vision entails a number of new research challenges that need to be solved before its potential can be fully exploited. In the remaining part of this paper, we outline three of the most peculiar ones, together with our results in addressing them within the PEIS-Ecology project.

## III. DEALING WITH HETEROGENEITY

The first fundamental challenge is heterogeneity. A PEIS-Ecology may include highly heterogeneous devices, which

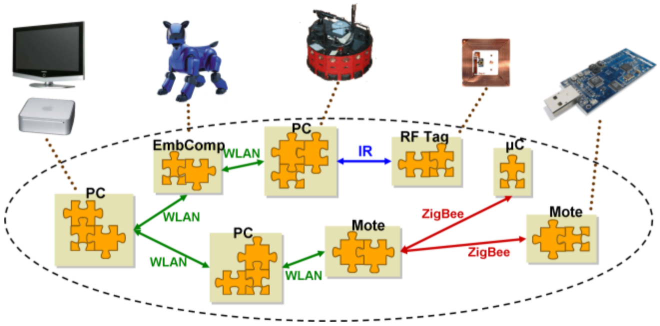


Fig. 2. A PEIS-Ecology may be highly heterogeneous in terms of hardware and software platforms, and of communication media.

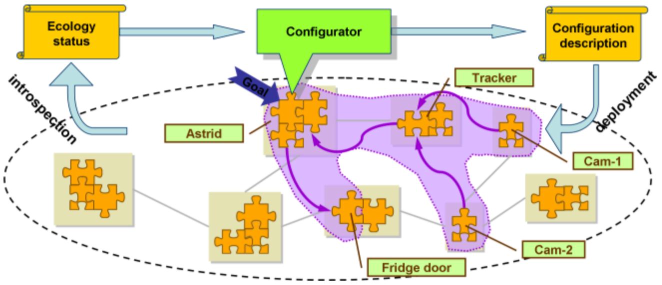


Fig. 3. The self-configuration problem in a PEIS-Ecology.

rely on different hardware and software platforms and different communication media — see Figure 2. Heterogeneity may also arise from the different levels at which the devices need to exchange information: from raw data streams to single measurement or to symbolic communication. In face of this, a PEIS-Ecology should provide the means to establish a meaningful communication between different PEIS. Physical diversity should be abstracted, and contents should make reference to a common ontology and measurement system. Achieving this requires a suitable middleware.

In our work, we have developed an open-source middleware, called the PEIS-Ecology middleware [8]. This middleware realizes the above PEIS-Ecology abstraction as follows.

The *uniform notion* of PEIS is implemented by providing a specialized component, called `peisinit`, which is run on each PEIS. The `peisinit` manages all the components which are physically present on that PEIS. In particular, it publishes advertisements that contain a semantic description of each component, it starts and stops components on request, it monitors the execution of each component, and it publishes a `fail` signal if a component fails.

The *uniform communication model* is implemented in the PEIS-kernel, a shared library which is linked by any component in any PEIS. The PEIS-kernel creates and maintains a fully decentralized P2P network, and performs services like network discovery and dynamic routing of messages between PEIS. The PEIS-kernel hides the underlying network heterogeneity, and it can cope with an environment in which PEIS and their individual components may appear and disappear at any time.

The *uniform cooperation model* is also implemented via the PEIS-kernel, which provides a distributed tuple-space augmented by event-based primitives `subscribe` and `unsubscribe`. If a component $A$ wants to use a functionality from a component $B$, possibly residing in a different PEIS, it subscribes to a suitable tuple key. When an insert operation is performed by $B$ with that key, component $A$ is notified and can consume the corresponding data. The subscription links are themselves represented by tuples in the tuple-space, called *meta-tuple*, thus allowing for both introspection and dynamic management of the subscriptions. As we discuss below, this feature is pivotal to the ability of the components in a PEIS-Ecology to autonomously create or change their cooperation pattern depending on the context.

The PEIS-kernel is highly portable, and it has been tested on hardware platforms ranging from multi-core 64 bit processors down to 8-bit microcontrollers, and on software environments such as Linux, MacOS, Windows, TinyOS, OpenR, and embedded Linux [9]. Moreover, the PEIS-middleware allows the inclusion into the PEIS-Ecology of very simple devices and everyday objects (e.g., an RFID-tagged coffee cup) using a "proxy" mechanism [10].

The full PEIS-Ecology middleware, including the PEIS-kernel, has been released as open-source under a set of GNU licenses, and it is available from the project website [5].

## IV. SELF-CONFIGURATION

Perhaps the strongest added value of a PEIS-Ecology comes from the ability to integrate the functionalities available in the different PEIS according to a given configuration, and to automatically create and modify this configuration depending on the current context. Here, the relevant contextual conditions include the current task(s), the state of the environment, and the resources available in the ecology. Self-configuration is the key to flexibility, adaptability and robustness of the system — in one word, to its *autonomy*.

The problem of self-configuration is a hard open problem for autonomous systems in general, and for distributed robotic systems in particular. Although much work has been done in several fields on the principles of self-configuration (e.g., ambient intelligence [11], web service composition [12], distributed middleware [13], autonomic computing [14]), no satisfactory solution exists. In a PEIS-Ecology, this problem is exacerbated by the fact that a PEIS-Ecology is highly heterogeneous and intrinsically dynamic.

Figure 3 illustrates the self-configuration problem in a PEIS-Ecology. The self-configuration process can be initiated by any PEIS whenever it needs to perform a task that may benefit from the help of other PEIS. In the example in the figure, this PEIS is the robot Astrid. In general, the configuration process operates by the following four steps [15], which are partly inspired by work in the field of web service composition.

1) assess the current state of the ecology (introspection), e.g., which functionalities are available and where;

2) generate a suitable configuration for the target task;
3) instantiate this configuration on the ecology by setting the needed parameters and establishing the needed subscriptions (deployment); and
4) monitor the execution of the generated configuration in order to detect possible failures and to start a re-configuration if needed.

In our realization of a PEIS-Ecology, step (1) is done using a combination of an *advertising mechanism*, that allows any PEIS to let all the other PEIS know about the functionalities it can provide; and a *discovery mechanism*, that allows each PEIS to find which other PEIS can provide a functionality compatible with its needs. The advertisement mechanism is implemented inside the `peisinit` component mentioned before.

Step (2) is the core step, and it can be implemented using different strategies. In our project, we have explored two complementary approaches for that. The first approach is a *plan-based*, centralized approach [16], [17]. In this approach, we use a global hierarchical planner to generate the (minimum cost) configuration for a given task. The second is a *reactive*, distributed approach [15]. In this approach, the configurator creates a local configuration, and assumes that the connected PEIS are able to recursively extend this configuration if needed. If they are not, the configurator receives a `fail` signal and tries a different local configuration.

The two approaches to configuration generation have the typical complementary strengths and weaknesses of plan-based and reactive approaches. The plan-based approach is guaranteed to find the optimal configuration if it exists, but it has problems to scale up and it cannot easily cope with changes in the ecology. The reactive approach scales up smoothly and it can quickly adapt to changes in the state of the ecology, but it might generate non-optimal configurations and it might fail to find a configuration even if one exists. Eventually, we hope to be able to combine these two approaches into a hybrid configurator.

Step (3) is done by exploiting the services of the PEIS-Kernel, in particular its ability to dynamically activate and parameterize components, and to create and destroy subscriptions between components via meta-tuples.

Step (4) relies on the monitoring capability of the `peisinit` which will issue a `fail` tuple whenever a component in the corresponding PEIS fails. In addition, some of the components may be able to detect their own failure (e.g., by detecting that they are operating outside their boundary conditions) and issue a corresponding `fail` tuple. The configuration process subscribes to all such tuples, and it tries to generate an alternative configuration if a `fail` signal is received.

## V. SHARING KNOWLEDGE

The last fundamental challenge of a PEIS-Ecology that we want to mention can be described as follows. In a classical robotic system, the robot's interaction with the environment and its objects is physically mediated: properties of the objects are estimated using sensors, and their state can be modified using actuators. In a PEIS-Ecology, a robot (PEIS) can also interact with an object (another PEIS) digitally: the robot can directly query properties from the object, and it can ask it to perform an action. Moreover, both digital and physical interaction can be mediated by other PEIS: e.g., another robot may observe the object and communicate the observed properties. The new challenge here is how to coordinate and integrate heterogeneous items of information that originate in both forms of interaction.

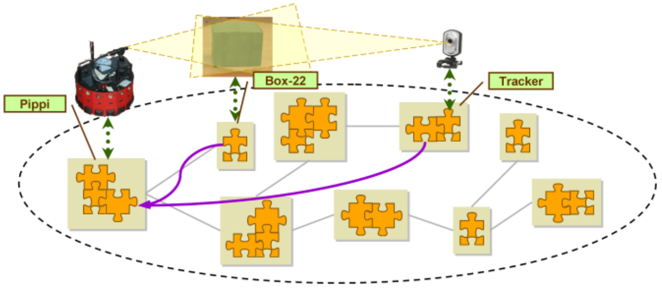


Fig. 4. The cooperative anchoring problem in a PEIS-Ecology.

Consider the case depicted in Figure 4. The robot Pippi is observing a box, and needs to know its position and contents. Assume that the box is a PEIS, with ID = `Peis301`, containing a Mote. If Pippi could establish that the physical box in front of it is the same object as `Peis301`, then it could ask the property 'content' from it by digital communication. Moreover, if Pippi can establish that the object which is observed by the ceiling camera is the same as the box in front of it, it could obtain a more precise estimate of the box' position.

Our approach to cope with this challenge is based on an extension of the concept of *perceptual anchoring* [18]. Anchoring is the process of connecting, inside an intelligent system, the symbols used to denote an object (e.g., `box-22`) and the percepts originating from the same objects (e.g., a green blob in the camera image). Our extension of anchoring to a cooperative setting is called *cooperative anchoring*.

We have developed a general framework for matching and fusing different types of information obtained from various sources in a PEIS-Ecology, intended as the basic underlying substrate needed to perform cooperative anchoring [19]. The work is inspired by our previous work on perceptual anchoring, and by the theory of conceptual spaces proposed by Peter Gärdenförs [20]. As such, it is well suited for dealing with both perceptual and non-perceptual (often symbolic) information. In particular, the main interest of the framework is that it allows the many different types of information which are present in a PEIS-Ecology to be efficiently represented, matched and fused using a single conceptual space representation, called *anchoring spaces*.

In the above example, Pippi represents its perceptual information (perceived color and shape of the box) in the shared anchoring space. It then queries the tuple-space for all `PhysicalRepresentation` tuples of each PEIS in the ecology (each PEIS must publish this tuple by convention).

(a) (b) (c) (d)

Fig. 5. Four views of the PEIS-Home experimental test-bed. (a) The kitchen, with the nose-equipped robot Pippi inspecting a smart fridge. (b) The living room, with the human-interface robot Astrid at its parking position. (c) A Mote-IV mote monitoring temperature and humidity of a plant. (d) The PEIS-Home simulation environment, in top-view.

All the received tuples are mapped to the shared anchoring space, and are matched with the perceived properties of the box. If the matching succeeds for a given PEIS, say the one with ID = `Peis301`, then Pippi can establish a correspondence between its local name `box-22` and `Peis301`. It can then ask additional properties to `Peis301` (e.g., its contents) and combine these properties with the observed ones. A similar matching and fusion process can be done to combine the perceptual information of Pippi with the one acquired by the ceiling camera, again by mapping this information in the common anchoring space.

## VI. EXPERIMENTAL VALIDATION

The PEIS-Ecology project follows a methodology which is strongly experimental: principles and techniques are systematically evaluated on physical platforms, and the results are used to refine these principles and techniques.

### *A. The* PEIS*-Home*

In order to follow this methodology, we have built a physical test-bed facility called the PEIS-Home. This facility looks like a typical Swedish bachelor apartment (Figure 5). The PEIS-Home is equipped with a communication infrastructure and with a number of PEIS, including static cameras, mobile robots, multi-media devices, a moving table, sensor nodes (motes), a refrigerator equipped with gas sensors and an RFID reader, and many more. A mid-fidelity 3D simulation has also been implemented using Gazebo [21].

We have used this test-bed to run a large number of experiments, with several aims: to test our technical solution and measure our progress; to validate the effectiveness of the developed techniques; to evaluate the acceptability of the PEIS-Ecology concept to human users, with special attention to elderly people; and to demonstrate the PEIS-Ecology concept to the society at large. In addition, we have implemented a version of the PEIS-Ecology on a second test-bed facility available at ETRI (Korea), and we have run a series of experiment on that test-bed. This second series of experiments has demonstrated the generality and portability of the PEIS-Ecology concept and middleware. Descriptions and videos of some of the experiments can be found on the PEIS-Ecology home page [5].

### *B. An illustrative experiment*

To give a concrete feeling of how the implemented PEIS-Ecology works, we report an experiment that involves all of the issues discussed in this paper. The experiment was the final demonstration of the PEIS-Ecology project.

**Experimental setup:** The experiment was run inside the PEIS-Home demonstrator. The following PEIS were used:

- Astrid, a PeopleBot robot from ActivMedia, equipped with a PTZ camera and a laser scanner; it includes components for task and configuration planning, navigation, wheel control, object recognition, and localization based on scan-matching.
- Tracker, a tracking system connected to a stereo camera mounted on the ceiling, capable of tracking multiple persons [22]; the tracker is also capable of tracking the position of Astrid.
- Table, a table equipped with an RFID tag reader.
- Lamp, a lamp controlled by a MoveIV T-Mote.
- HSM, the 'Home Security Monitor', a PC running a monitoring component, a phone interface, and a ZigBee bridge.

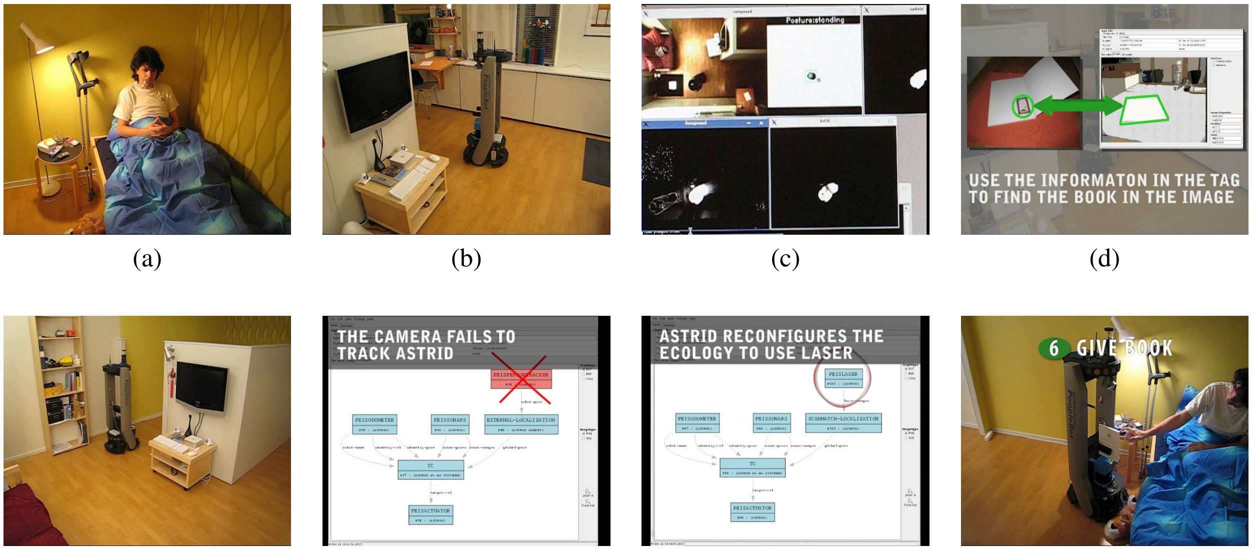
Several other PEIS were also active in the PEIS-Home, but they are not relevant here. All the PEIS run the PEIS-kernel, and they are all connected via 802.11 WLAN, except the Lamp which is connected via ZigBee.

**Execution:** The experiment unrolls as follows. Snapshots from some of the steps are shown in Figure 6.

1) Alex is laying in bed with a broken leg, and he asks the PEIS-Ecology to bring him a given book using a mobile phone interface. The request is received by the HSM, which forwards the task to Astrid.
2) Astrid queries the tuple-space to retrieve the current state of the ecology, including the position of the book. It finds the tuple giving `table` for the book position, published by the Table which had previously detected the RFID tag of the book.
3) Astrid generates the action plan:[2] (turnon light), (goto table), (anchor book1), (grasp book1), (goto bed), (deliver book). The first action is needed for reliable navigation.

---

[2] The plan has been slightly condensed for explanation purposes.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. Some snapshots from the execution of the "fetch book" experiment. (a) Alex is ill. (b) Astrid navigates to the table. (c) The Tracker tracks the position of Astrid. (d) Astrid anchors the book using information provided by Table. (e) Astrid enters the bedroom. (f) The Tracker fails. (g) An alternative configuration is found. (h) task completed. The full video of this experiment can be accessed from the project website [5].

4) Astrid generates a configuration to execute the action (turnon light); this configuration only involves activating the Light PEIS with the parameter `on`.
5) Astrid generates a configuration to execute the action (goto table); this configuration includes the activation of the Tracker and the connection of its output to Astrid's navigation component.
6) Astrid generates a configuration to execute the action (anchor book1). This includes the Table, which reads the properties in the book's RFID tag (e.g., the color of its cover) and Astrid's camera, in order to identify which of the objects on the table is `book1`.
7) Astrid grasps the book (here, with the help of a human because of the limited manipulation abilities of the PeopleBot) and generates a configuration for (goto bed), again involving the output from the Tracker.
8) When Astrid enters the bedroom, the Tracker fails to track it because of its limited field of view and it sends a `fail` tuple. Astrid then searches an alternative configuration for the action (goto bed), and finds one using its on-board laser plus scan-matcher for self-localization.[3]
9) Astrid reaches the bed and delivers the book.

**Discussion:** Although the task was relatively simple, this experiment illustrates most of the points discussed in the previous sections. Step 2 shows that the tuple-space enables the sharing of information from different sources. It also shows that simple objects can be seen as part of the PEIS-Ecology using the "proxy" mechanism in the PEIS-middleware. Step 4 shows that the PEIS-Kernel allows the seamless integration of computationally rich devices, like the robot Astrid, and computationally poor devices, like the mote controlling the lamp. It also shows how the P2P network created by the PEIS-Kernel spans heterogeneous physical networks: the WLAN used by Astrid and the ZigBee network used by the mote. Step 5 shows the dynamic generation and deployment of a PEIS-Ecology configuration using the configuration planner. Planning was done using an efficient HTN planner [16], so it only required a fraction of a second. Step 6 shows a concrete example of anchoring digital and perceptual information. Finally, Step 8 shows a case of recovery from failure through dynamic re-configuration.

## VII. CONCLUSIONS AND OUTLOOK

The idea to integrate robots and smart environments is starting to pop up at several places and under several names, including network robot systems [23], intelligent space [2], sensor-actuator networks [3], ubiquitous robotics [4], artificial ecosystems [13], and still others. A few projects were recently started with the aim to explore the scientific, technological and practical implications of this integration. Currently the largest efforts are probably the Network Robot Forum [1], the U-RT project at AIST [24], and the Korean Ubiquitous Robot Companion program [4]. The PEIS-Ecology project presented in this paper is part of the latter effort. This project is distinct in its emphasis on study of the fundamental scientific principles that underlie the design and operation of a highly distributed, highly heterogeneous robotic system.

The PEIS-Ecology project has achieved a number of scientific and technological results, including the three ones reported in this note: the development of a flexible and light-weight open-source middleware, the development of techniques for self-configuration and re-configuration, and the study of the integration between physical and digital

---

[3] The laser was not selected in the first place because of its greater cost in term of energy.

interaction in a distributed robot system. Source code, videos and scientific papers relative to these achievements are available at the PEIS-Ecology web site [5].

The next important step in this development will be the inclusion of humans into a PEIS-Ecology. In the PEIS-Ecology vision, humans can be seen as just another species that operate in symbiosis with the robotic devices in the ecology [25]. The PEIS-Ecology should be able to infer the status and intentions of humans from observations, and adapt its behavior to that. For instance, if a human shows the intention to relax, the vacuum cleaner should move to a different room. A PEIS-Ecology should also be able to infer what the humans can afford to it: for instance, the vacuum cleaner could ask the human to empty its dust-bag if it knows that the human can afford that. Ideally, it should also be able to smoothly update its model of what a human user can afford to adapt to changes in this user, e.g., growing older. Although some preliminary steps have been made toward the inclusion of humans in a PEIS-Ecology [26], [27], [28], this issue remains largely open for future exploration.

## ACKNOWELDGEMENTS

This work was supported by ETRI (Electronic and Telecommunications Research Institute, Korea) through the project "Embedded Component Technology and Standardization for URC (2004-2008)", by CUGS (the Swedish National Graduate School in Computer Science), and by Vetenskapsrådet (the Swedish Research Council). Many thanks to all the people at the AASS Mobile Robotics Lab of Örebro University, Sweden, for their contributions to this project.

## REFERENCES

[1] "Network Robot Forum," www.scat.or.jp/nrf/English/.

[2] J. Lee and H. Hashimoto, "Intelligent space – concept and contents," *Advanced Robotics*, vol. 16, no. 3, pp. 265–280, 2002.

[3] F. Dressler, "Self-organization in autonomous sensor/actuator networks," in *Proc of the 19th IEEE Int Conf on Architecture of Computing Systems*, 2006.

[4] J. Kim, Y. Kim, and K. Lee, "The third generation of robotics: Ubiquitous robot," in *Proc of the 2nd Int Conf on Autonomous Robots and Agents*, Palmerston North, New Zealand, 2004.

[5] "The PEIS ecology project," Official web site, www.aass.oru.se/˜peis/.

[6] A. Saffiotti and M. Broxvall, "PEIS ecologies: Ambient intelligence meets autonomous robotics," in *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, Grenoble, France, 2005, pp. 275–280.

[7] M. Broxvall, S. Coradeschi, A. Loutfi, and A. Saffiotti, "An ecological approach to odour recognition in intelligent environments," in *Proc of the IEEE Int Conf on Robotics and Automation*, Orlando, FL, 2006.

[8] M. Broxvall, M. Gritti, A. Saffiotti, B. Seo, and Y. Cho, "PEIS ecology: Integrating robots into smart environments," in *Proc of the IEEE Int Conf on Robotics and Automation*, Orlando, FL, 2006.

[9] M. Bordignon, J. Rashid, M. Broxvall, and A. Saffiotti, "Seamless integration of robots and tiny embedded devices in a peis-ecology," in *Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, San Diego, CA, 2007.

[10] J. Rashid, M. Broxvall, and A. Safffiotti, "Digital representation of everyday objects in a robot ecology via proxies," in *Proc of the IEEE/RSJ Int Con on Intelligent Robots and Systems*, 2008.

[11] A. Kaminsky, "Infrastructure for distributed applications in ad hoc networks of small mobile wireless devices," Rochester Institute of Technology, IT Lab, Tech. Rep., May 2001.

[12] J. Rao and X. Su, "A survey of automated web service composition methods," in *Proc of the 1st Int Workshop on Semantic Web Services and Web Process Composition*, San Diego, California, USA, 2004.

[13] A. Sgorbissa and R. Zaccaria, "The artificial ecosystem: a distributed approach to service robotics," in *Proc of the IEEE Int Conf on Robotics and Automation (ICRA)*, 2004.

[14] G. Tesauro, D. Chess, W. Walsh, R. Das, A. Segal, I. Whalley, J. Kephart, and S. White, "A multi-agent systems approach to autonomic computing," in *Proc of the Int Conf on Autonomous Agents and Multiagent Systems*, 2004, pp. 464–471.

[15] M. Gritti, M. Broxvall, and A. Saffiotti, "Reactive self-configuration of an ecology of robots," in *Proc of the ICRA-07 Workshop on Network Robot Systems*, Rome, Italy, 2007.

[16] R. Lundh, L. Karlsson, and A. Saffiotti, "Plan-based configuration of an ecology of robots," in *Proc of the IEEE Int Conf on Robotics and Automation*, Rome, Italy, 2007.

[17] ——, "Autonomous functional configuration of a network robot system," *Robotics and Autonomous Systems*, 2008, in press.

[18] S. Coradeschi and A. Saffiotti, "An introduction to the anchoring problem," *Robotics and Autonomous Systems*, vol. 43, no. 2-3, pp. 85–96, 2003.

[19] K. LeBlanc and A. Saffiotti, "Cooperative anchoring in heterogeneous multi-robot systems," in *Proc of the IEEE Int Conf on Robotics and Automation (ICRA)*, Pasadena, CA, 2008.

[20] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*. Cambridge, MA, USA: MIT Press, 2000.

[21] "Player/Stage/Gazebo website," http://player.sourceforge.net.

[22] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente, "People detection and tracking using stereo vision and color," *Image and Vision Computing*, vol. 25, no. 6, pp. 995–1007, 2007.

[23] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Special issue on network robot systems," *Robotics and Autonomous Systems*, 2008, in press.

[24] O. Lemaire, K. Ohba, and S. Hirai, "Dynamic integration of ubiquitous robotic systems using ontologies and the rt middleware," in *Proc of the 3rd Int Conf on Ubiquitous Robots and Ambient Intelligence*, Seoul, Korea, 2006.

[25] S. Coradeschi and A. Saffiotti, "Symbiotic robotic systems: Humans, robots, and smart environments," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 82–84, 2006, special issue on the future of AI.

[26] M. Broxvall, A. Loutfi, and A. Saffiotti, "Interacting with a robot ecology using task templates," in *Proc. of the 16th IEEE Int Symp on Robot and Human Interactive Communication (RoMan)*, Jeju Island, Korea, 2007.

[27] D. Guarino and A. Saffiotti, "Using fuzzy logic to monitor the state of an ubiquitous robotic system," *Journal of Uncertain Systems*, vol. 2, no. 2, pp. 121–132, 2008.

[28] M. Cirillo, L. Karlsson, and A. Saffiotti, "A framework for human-aware robot planning," in *Proc of the Scandinavian Conference on Artificial Intelligence*, 2008.