Dynamic Speech Interaction for Robotic Agents

Jan Koch, Holger Jung, Jens Wettach, Geza Nemeth and Karsten Berns

Robotics Research Lab
University of Kaiserslautern
Gottlieb-Daimler-Straße, 67663 Kaiserslautern, Germany
{koch, h_jung, wettach, berns}@informatik.uni-kl.de

Speech Technology Laboratory
Budapest University of Technology and Economics
1117. Budapest, Magyar tudosok krt.2., Hungary
nemeth@tmit.bme.hu

***Abstract*— Research in mobile service robotics aims on development of intuitive speech interfaces for human-robot interaction. We see a service robot as a part of an intelligent environment and want to step forward discussing a concept where a robot does not only offer its own features via natural speech interaction but also becomes a transactive agent featuring other services' interfaces. The provided framework makes provisions for the dynamic registration of speech interfaces to allow a loosely-coupled flexible and scalable environment. An intelligent environment can evolve out of multimedia devices, home automation, communication, security, and emergency technology. These appliances offer typical wireless or stationary control interfaces. The number of different control paradigms and differently lay-outed control devices gives a certain border in usability. As speech interfaces offer a more natural way to interact intuitively with technology we propose to centralize a general speech engine on a robotic unit. This has two reasons: The acceptance to talk to a mobile unit is estimated to be higher rather than to talk to an ambient system where no communication partner is visible. Additionally the devices or functionalities to be controlled in most cases do not provide a speech interface but offer only proprietary access.**

***Index Terms*— HCI, mobile robots, transactive systems, ambient intelligence**

I. Introduction

To access different types of hardware devices and information technology, people have to learn to deal with a lot of proprietary interfaces. In most times this is related to remembering which buttons to press on the devices. We envision highly ambient technology in home environments with lots of information and functionality to offer. Human speech especially for elderly people seems to be a better choice to interact with technology. Though generally the quality of speech recognition is not perfect yet, the border of usability is lower as with pressing buttons because a lot of elderly people are scared to come in touch with technology. We propose a generalized concept for speech integration on a mobile robotic unit. The robot serves as interaction partner as it seems more convenient to talk to a certain visible autonomous device, like a pet, than to a light switch for example. Generally mobile robots offer great opportunities in the area of human-machine interaction [1]. When integrated into the home environment the robot becomes a transactive agent. It processes control and information transfer between human users and ambient technology. Controllable and information sharing devices register their speech interface dynamically at a dialog engine which runs on the robot. Users this way are able to use the robot for interaction with systems that by default do not have the capability to offer speech interfaces. The robot this way does not only become interactive, it becomes transactive as it is relaying the control information to other components and responds in their role. The mobile robot additionally offers several features regarding human recognition to improve the audio-related quality of speech recognition in particular.

The speech engine, the mobile unit's characteristics as well as the implemented appliances are presented within this work. In the next chapter we start with an overview on development and examples of speech interaction on mobile systems and in intelligent environments. The robot prototypes that currently have our speech engine running are introduced in III. Our approach to implement a dynamic dialog system is explained in IV. Furthermore in chapter V we give some examples on applications that have been facilitated using the speech engine on the robots and discuss conclusions in VI.

II. Speech Interaction with Service Robots and Intelligent Environments

There are several important reasons to use man-machine dialog systems based on spoken language. Input by speech can be faster than with haptive hardware while hands and eyes are free to focus on other activities. Especially for older people or people who lack experience in computer usage, it is the simplest way to interact with technology by speech.

Another advantage is the independence from the distance between user and machine when using room microphones. It is even possible to access the system remotely via telephone or internet. Today, there are many areas where speech systems already are used. The most important areas are:

- home automation (lighting, entertainment electronic, answerphone),
- industrial use (control of machines, e.g. for quality checks or dispatch),
- office use (database queries, dictation, organisation of documents),
- banking systems (bank assignments, stockjobbing, credit cards),
- public transport and other information services (time table informations, booking, weather forecast, event notes),
- medical appliances (diagnoses systems, interphone systems for patients, surgeon aids),
- elderly and disabled people (emergency systems),

There are currently several projects from industry, institutes and universities where dialog systems are developed. Many projects are developed for simple database queries. A big research project is Smartkom [2], a multimodal task-oriented dialog system for two main domains, information seeking in databases and device control. To improve the acceptance and to make control easier, dialog systems have also been implemented on many robotic systems. Usually, dialog systems are used here to offer controls for easy tasks, as to use the robot as information systems or to get information about its technical status. Examples are CARL from the University of Aachen [3] or HERMES from the University of Munich [4]. Both can perform several easy jobs when the user asks for it. Armar from Karlsruhe [5] and another robot called CARL from Aveiro, Portugal [6] can fulfil kitchen tasks. The Japanese robots Jijo-2 [7] and HRP-2 [8] can be asked for people, for their room numbers or for the path to their location within an office building. At the University of Bielefeld, the robot BIRON [9] is developed which can learn the name of artifacts if the user shows them. There is also a current research at the University of Saarbruecken [10], where a system is developed which tries to recognize the intention the user has.

All these projects are application-driven and the robot is optimized to perform certain tasks while providing a suitable specific speech interface. Our approach goes into the opposite direction where the robot only has rudimentary functionality but behaves as a transactive system, offering control and information of various other parts of the ambient environmet over a unified speech interface engine. It has to be mentioned that in this work the improvement of the quality of speech recognition or synthesis was not a subject. The idea is to keep the dialog system independent from the implementation of recognizer and synthesizer. Nevertheless the libraries we selected to perform these task in our examples are described in section IV.

Multiple research groups address the field of intelligent indoor environments. The intelligent room project at MIT uses architectural ideas of the area of mobile robotics [11] and focuses on supporting humans by interpretation of their actions, gestures, and speech. The Amigo-project in Germany [12], the Aware Home in Atlanta or the I-Living project are also examples for intelligent environments which shall serve humans in their daily life.

III. Robot Platform

In this work the mobile prototypes MARVIN[1] and ARTOS carry speech capabilities in terms of our speech engine. MARVIN is a experimental platform for basic research on indoor exploration and human detection. ARTOS is has been optimized in size and power consumption for usage in home environments.

Fig. 1. MARVIN

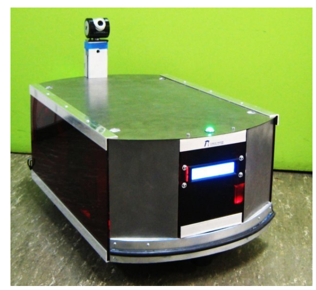

Fig. 2. ARTOS

A. MARVIN

MARVIN (Mobile Autonomous Robot Vehicle for Indoor Navigation, see fig.1) is a test platform for autonomous exploration of indoor environments, 3D mapping and environmental modelling. The basic vehicle concept consists of a differential drive and two planar laser range scanners for obstacle detection and SLAM. The robot control system is implemented as behaviour-based approach on several levels of abstraction beginning on basic collision avoidance and trajectory control, up to exploration and mapping. In this context a behavioral network has been developed that enables the robot to explore a structured, but a-priori unknown indoor environment without any user intervention and to create a topological map of all visited rooms and their connections (see [13]).

B. ARTOS

ARTOS (Autonomous Robot for Transport and Service) is under development for the BelAmI project[2]. Within that

[1] **http://www.agrosy.informatik.uni-kl.de**
[2] **http://www.belami-project.de**

TABLE I

SOME FACTS ABOUT ARTOS

Weight	20kg
Height, Width, Length	25cm, 30cm, 50cm
Max Speed	60cm per sec
Sensors	Laser, Infrared, Ultrasonic, Bumper
Multimedia	Speakers, Micro, CCD Camera
Navigation	Odometry, RFID Landmarks
Operation time	6h at 40W
Control Framework	MCA, behavior-based collision avoidance

project, an assisted living environment for elderly care is assembled. ARTOS shall serves three purposes:

- Transport aid for items of the daily life to help disabled people
- Teleoperated or autonomous emergency recognition
- Multimedia and system interaction agent

ARTOS uses an Hokuyo laser range finder, ultrasonic and infrared sensors for collision avoidance. The Modular Controller Architecture (MCA), see section IV-E, is used as control framework. For the navigational subsystems a behavior-based control approach has been chosen as we use it on all of our robots [14]. Artos is controlled by a low-power PC so that the overall power consumption with all sensors and multimedia activated at maximum drive speed does not exceed 40W. An overview of the technical details is given in Table I.

As the interaction aspect is the main topic of this paper we will continue with a description of the speech engine and return to the robot and the implemented scenarios in section V.

IV. THE DYNAMIC SPEECH SYSTEM

As motivated in the introduction a mobile robot should be enabled to interact and transact providing its own functionalities and those of other services to human inhabitants of intelligent environments. From the technical point of view the service-interaction is negotiated as follows: An application or the robot itself hands over the dialog description to the speech engine. The dialog is registered and set active. While traversing the dialog the engine calls back the registering application, passing state variables that might have been changed by the human during interaction. Finally dialogs may terminate or be unregistered. The dialog engine therefore consists of several parts:

- Registration and deregistration of dialogs
- Parser for dialog files
- Traversing and processing dialog representations
- Callback to registering application

A coarse overview of the system is given in Fig. 3.

A. Dialog Specification

Dialogs are specified in XML using a syntax that has been developed in [15]. The scheme includes elements for describing speech inputs and outputs, to specify speech parameters,

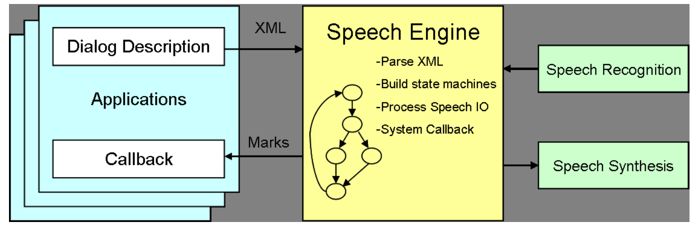

Fig. 3. Architecture of Dynamic Speech System

and to define condition and value variables for controlling and adapting the dialog flow. Upon dialog registration the given XML is parsed and mapped to state machines.

The settings part of a dialog specification contains parameters for influencing the behavior of the speech synthesis library. Special actions can be globally defined which are engaged if the user inputs some prescribed words or makes no input for a certain time.

In the main dialog section user inputs which have to be recognized by the speech recognition system and machine outputs which have to be performed by the speech synthesis are specified. Other elements define conditions which have to be fulfilled, e.g. using variables and corresponding values or by comparing two variables. therefore variables can be declared with a certain type and their value can be set initially and be changed during runtime by the user or the machine. Also an element to stop the control flow for a certain time is defined; jumps in the dialog flow are also possible. A very important element is defined for the interaction between the dialog engine and the application that registered the respective dialog.

The example presented in Fig. IV-A is a simple dialog where a user can ask the robot ARTOS about foods comprised in an intelligent refrigerator. In the element `waiting_for_userinput` beginning in line 1, it is possible to design some `user_input` elements. Each of these elements begins with the element `sentence`. The sentence defined here has to be recognized by the speech recognition system, then elements inside of that special `user_input` element are processed. With these `user_input` elements inside of a `waiting_for_userinput` element, it is possible to define branches in the dialog flow depending of a users speech input. At the end of the `waiting_for_userinput` element, the branches are merged together again. Here in the first example, there is only one possibility to make a speech input. If the sentence "ARTOS, which food items are in my fridge?" is asked by user, the element `prompt` is affected. Here it is possible to define a sentence to be spoken by the speech synthesis system. The sign "%" marks a variable, the system substitutes automatically with it is values. This will be shown in more detail in the second example.

The second example in fig.IV-A describes a more complex dialog. Here the system triggers an output if there are some food items in the fridge which are expired. therefore at the beginning, two variables are defined, one, `itemCount`, concludes the number of expired foods. Its default value is zero. The other variable, `itemNames`, contains the food items' names. Beginning in line four, a condition depending on the value of `itemCount` is defined. If its value is zero, there are no expired foods and nothing is to do. Otherwise if the value is higher than zero, it is necessary distinguish again between singular and plural for building a grammatically correct sentence. Depending on these inputs, a sentence is generated by the speech synthesis library. As already mentioned, the variables marked with "%" are substituted by their values. If the system sets the food variables, one of the two conditions in line four and five becomes true and the corresponding sentence is synthesized. In general it is not necessary to define sentences here, any other XML dialog element could also be used inside of the condition elements.

After handling the conditions, the variables are set back to their default values. Then, there is a jump back in the dialog flow from line nine to three and the dialog system waits again if variables are externally set and one of the two conditions becomes true.

B. Synthesis

We are using the speech synthesis engine *Profivox* developed at the Speech Technology Laboratory at Budapest University of Technology and Economics [16], [17]. Profivox is under development since 1997. Its goals are intelligible human-like voice, robust software technology for continuous running, automatic conversion of declarative sentences and questions, the possibility of tuning for application-oriented special demands and real-time parallel running on minimum 30 channels. Profivox is built up modular to make the system multilingual and easier to develop by simply changing single modules. A control module manages every step necessary to produce synthesized speech.

C. Recognition

The speech recognition engine *AmiASR* comes to use. It has also been developed at Budapest University of Technology. This software is based on the *Hidden Markov Model Toolkit* (HTK)[3] [18]. Input for the recognition system are a dictionary where all valid words are listed and a grammar file with the words connected to sentences to be detected. Generally spoken, a speech recognizer works better if the set of words to be recognized is limited to the necessary amount. In our system we therefore generate dictionary and grammar on the fly for each point in the dialog.

D. Dialog Engine

The speech engine becomes active when dialogs are registered. These are modelled in XML and are converted by the speech engine to states and transitions of state machines. Each dialog is represented by a state machine. Within states, speech output can be triggered. Transitions are traversed when condition variables are valid or the user has given matching speech input. All dialogs share a common start state. Three possibilities exist to walk into a certain dialog from here:

- Matching speech input
- Valid condition variable
- Manually triggered start

When reaching a state the engine at first generates and triggers the speech synthesis if corresponding output is specified for that state. Furthermore the dictionary and grammar files are produced containing the words and sentences whose match might be the condition for transitions emerging from this state. Afterwards the recognizer is activated if speech input is defined. Finally the transitions that require the fulfillment of conditions regarding dialog variables are checked.

The engine continues then to periodically check whether either a matching user input has been given or whether a condition has been fulfilled. Within the dialog specification also a timeout may be given that leads to a special transition. Also transitions for help or termination may be specified that are connected to each state of the dialog. If a dialog reaches its final state, the engine returns to the starting state.

[3] **http://htk.eng.cam.ac.uk**

```
<dialog dialog_id="main">
  <waiting_for_userinput>
    <user_input>
      <sentence>ARTOS, which food items are
                in my fridge?
      </sentence>
      <prompt>You have %items in your fridge.
      </prompt>
    </user_input>
  </waiting_for_userinput>
</dialog>
```

Fig. 4. Example 1 of a dialog definition

```
0 <dialog dialog_id="main">
1   <var type="int" name="itemCount" value="0"/>
2   <var type="string" name="items" value=""/>
3   <mark name="begin"/>
4   <if var="itemCount" value="1" relation="==">
      Consider that %itemNames has expired.
5     <elseif var="numberFoods" value="1"
       relation=">">
        Consider that %itemNames have expired.
      </elseif>
6   </if>
7   <set_variable name="itemNames" value=""/>
8   <set_variable name="itemCount" value="0"/>
9   <goto mark="begin"/>
10 </dialog>
```

Fig. 5. Example 2 of a dialog definition

During runtime, new files with dialogs can be added or files can be removed. Then new states and transitions are added to the finite state machine or states and transitions are removed. When an application removes a dialog the design decision has been to complete the currently running dialog and then remove it from the set of state machines.

E. System Integration and Callback

For communication between speech engine and application the Modular Controller Architecture (MCA) is used. MCA has been developed for mobile robotics research, initially at the FZI in Karlsruhe[4] [19] and is under continuous development[5]. MCA enables primitive and object-oriented communication, re-usability, component-based development, hardware access and application distribution. From the robot side the speech engine is accessed by those particular MCA modules that are designed to share data and control with a human user. When an ambient service wishes to register its interface it has to be written either in MCA, too, and this way perform a native connection or use one of the other middleware solutions that have been developed within the BelAmI project, see [20].

System call-backs are triggered by asynchronous messages. States of variables are communicated via the shared-memory concept of MCA. The XML representation for interaction dialogs is stored within the shared data structure by the registering application. A notification of the application triggers parsing and translation of the internal state machine representation of the dialog by the speech engine. If dialogs have been registered that require activation by the application another notification is necessary.

Applications provide marker variables within their XML dialog description as described in IV-A. The value of markers is changed when reaching certain states. While traversing the state machine the speech engine sets the markers to the according values. These may be boolean variables for simply reaching the dedicated states or string variables that are filled with default values or even dynamic string values that have been derived from human speech input. The dialog registering application is then notified and may access the state of its marker variable within the shared memory. In the same way the application may have specified variables that are subject to change during runtime. These variables are stored in a different shared memory.

We continue by showing how these mechanisms have been utilized to facilitate application scenarios.

V. Application Scenarios

The topological map, described as one feature of MARVIN in section III-A is used for a 'tourist guidance tour' through the RRLAB as demonstration application for speech-based human-machine interaction. This functionality in concept does not differ much from the well-known robotic museum tour guides that have been in use for some time now [21]. We improved the human detection however to enhance the quality of interaction fusing information from different sensors. That is, detection of sound sources with a stereo microphone system [22], of pairs of legs with the laser scanners and consequently of faces with a webcam mounted on a pan-tilt unit helps to perform the speech interaction more naturally. Hence the appearance of a human-being hypothesis in the robot's vicinity triggers the start of the dialog. The lab guidance tour is accomplished by approaching different outstanding positions in the robots map where the robot introduces itself, presents historical information about the research group and explains several actual and future research topics as well as its colleagues, see Fig. 6. To achieve this functionality a new behaviour component has been added to the higher level control system which interacts which the dialog control system to trigger speech synthesis, recognition and robot motion as required.

Fig. 6. MARVIN giving a tour through the RRLAB

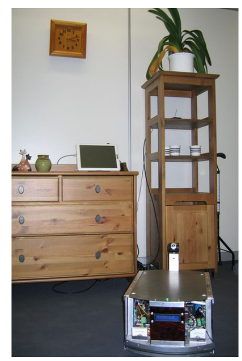

Fig. 7. ARTOS in the Assisted Living Laboratory

The robot ARTOS uses speech for giving reminder and system messages within the Assisted Living Lab at Fraunhofer IESE [23] [24]. The Assisted Living Lab is a fully equipped living area, see Fig. 7. Several dialogs may be registered at runtime as described in the following section. They are distinguished as human-triggered and system-triggered. Some examples are given as follows: We have an intelligent refrigerator that checks the expiry of food items using RFID. The dialog description, used as example in section IV-A represents this funcionality. Expiry warnings are presented to the human by the robot ARTOS. The refrigerator therefore has to register its dialogs at the speech engine of ARTOS. It consists of warning messages containing the food descriptions that are provided by the refrigerator as variable strings. As shown in Fig. IV-A the dialog differs if one ore more items have expired. Also the robot may be asked by the human about the refrigerator content.

Furthermore dialogs can be dynamically registered by a generic warning and reminder service that collects and

[4] **http://www.fzi.de/ids**

[5] **http://sourceforge.net/projects/mca2**

distributes information in our intelligent environment. Here the robot serves as true agent for the system, while all other examples are more or less related to its own functionality.

A lot of more applications however are planned to be integrated. A service for reminding elderly people to consume enough liquids and nutrition is already included in the assisted living area. Also a home automation system that controls all door sensors, light and shading switches is available. Two systems to gather the inhabitants location have been integrated. These systems will be added to the speech interface repository in the near future. For example we aspire simple transport functionalities for the robot as well as communication relaying. The human therefore may call the mobile unit to approach his location. Our indoor localisation system will register this interaction dialog at the robot.

VI. Conclusion and Outlook

We presented an engine that dynamically processes speech dialogs for human-machine interaction with transactive mobile robotic units in intelligent home environments. The dialogue machine is installed on the mobile robots ARTOS and MARVIN and is continuously used and tested within several applications and scenarios. Especially the possibility to dynamically add and remove dialogs turned out to be helpful for fast prototyping and testing as well as for the ambient character of our test environment.

Apart from the existing appliances a number of additional services will be included within the next months. Further work will be done regarding the speech recognition and synthesis quality. Problems with dialogs starting with the same condition variable or speech input triggers have to be addressed. Also the middleware connectivity that is currently built in MCA shall be migrated to a more open standard.

VII. Acknowledgements

We gratefully acknowledge the funding of the BelAmI project by the State Ministry of Science, Education, Research and Culture of Rhineland-Palatinate, by the Fraunhofer-Society and by the German Federal Ministry of Education and Research.

References

[1] A. Tapus, M. Matarić, and B. Scassellati, “The grand challenges in socially assistive robotics,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, March 2007.

[2] J. Alexandersson and T. Becker, “Overlay as the basic operation for discourse processing in a multimodal dialogue system,” in *2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Seattle, 2001.

[3] F. Dylla and G. Lakemeyer, “A speech interface for a mobile robot controlled by golog,” *CogRob-2000*, 2000.

[4] R. Bischoff and V. Graefe, “Dependable multimodal communication and interaction with robotic assistants,” *11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002.

[5] R. Stiefelhagen, C. Fugen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, “Natural human-robot interaction using speech, head pose and gestures,” *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2004.

[6] L. Lopes, “Carl: from situated activity to language level interaction and learning,” *Proceedings of the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2002.

[7] T. Matsui, H. Asoh, J. Fry, Y. Motomura, F. Asano, T. Kurita, I. Hara, and N. Otsu, “Integrated natural spoken dialogue system of jijo-2 mobile robot for office services,” in *Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, 1999.

[8] J. Ido, Y. Matsumoto, T. Ogasawara, and R. Nisimura, “Humanoid with interaction ability using vision and speech information,” *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[9] I. Toptsis, A. Haasch, S. Hüwel, J. Fritsch, and G. Fink, “Modality integration and dialog management for a robotic assistant,” *European Conference on Speech Communication and Technology*, 2005.

[10] S. Wilske and G. Kruijff, “Service robots dealing with indirect speech acts,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[11] R. Brooks, “The intelligent room project,” in *Second International Cognitive Technology Conference (CT)*, Aizu, Japan, 1997.

[12] C. Magerkurth, R. Etter, M. Janse, J. Kela, O. Kocsis, and F. Ramparany., “An intelligent user service architecture or networked home environments.” in *2nd International Conference on Intelligent Environments*, Athen, Greece, July 5-6 2006, pp. 361–370.

[13] D. Schmidt, T. Luksch, J. Wettach, and K. Berns, “Autonomous behavior-based exploration of office environments,” in *3rd International Conference on Informatics in Control, Automation and Robotics - ICINCO*, vol. Robotics and Automation, Setubal, Portugal, August 1-5 2006, pp. pp. 235–240.

[14] M. Proetzsch, T. Luksch, and K. Berns, “Behaviour-based motion control for offroad navigation,” in *International Workshop on Humanitarian Demining (HUDEM)*, Brussels, Belgium, June 16-18 2004.

[15] C. Hauck, “Conception and integration of speech services into an ambient intelligence platform,” 2006.

[16] G. Nemeth, G. Olaszy, P. Olaszi, G. Kiss, C. Zainko, and G. Gordos, “Profivox - a hungarian text-to-speech system for telecommunications applications,” *International Journal of Speech Technology*, vol. 3, pp. pp. 201–215, 2000.

[17] I. Koutny, G. Olaszy, and P. Olaszi, “Prosody prediction from text in hungarian and it’s realization in tts conversion,” *International Journal of Speech Technology*, vol. 3, 2000.

[18] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, J. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Microsoft Corporated, 2000.

[19] K. U. Scholl, J. Albiez, and G. Gassmann, “Mca- an expandable modular controller architecture,” in *3rd Real-Time Linux Workshop*, Milano, Italy, 2001.

[20] M. Anastasopoulos, H. Klus, J. Koch, D. Niebuhr, and E. Werkman, “Doami - a middleware platform facilitating (re-)configuration in ubiquitous systems,” in *International Workshop on System Support for Ubiquitous Computing (UbiSys) at the 8th International Conference of Ubiquitous Computing (Ubicomp),* New Port Beach, California, USA, 2006.

[21] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, “The interactive museum tour-guide robot,” *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, July 26-30 1998.

[22] J. Lantto, “Sound source detection system for control of an autonomous mobile robot, a behaviour-based approach,” in *16-th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control (ROMANSY)*, 2006.

[23] J. Nehmer, A. Karshmer, M. Becker, and R. Lamm, “Living assistance systems - an ambient intelligence approach,” in *Proceedings of the 28th International Conference on Software Engineering (ICSE)*, Shanghai, China, May 20-28 2006.

[24] M. Becker, E. Ras, and J. Koch, “Engineering tele-health solutions in the ambient assisted living lab,” in *21st International Conference on 21st International Conference on Advanced Information Networking and Applications (AINA)*, Niagara Falls, Canada, 2007.