LTPC 2012

Proceedings of the

First International Workshop on Language Technology in Pervasive Computing (LTPC)

held in conjunction with PERVASIVE 2012 Conference

at Newcastle University, UK

June 19, 2012

German Research Center for Artificial Intelligence GmbH

SFB/TR 8 SPATIAL COGNITION

Organizers

Christoph Stahl (DFKI GmbH, Bremen)

Dimitra Anastasiou (University of Bremen)

Programme Committee

Michael Feld (DFKI, Saarbrücken)

Lars Bo Larsen (Aalborg University)

Nikolaos Mavridis (MIT and United Arab Emirates University)

Mice McTear (University of Ulster)

Norbert Pfleger (SemVox, Saarbrücken)

Contact Address

Dr. Christoph Stahl
DFKI GmbH
Cyber-Physical Systems
Cartesium, Room 0.59
Enrique-Schmidt Straße 5
28359 Bremen
Germany

phone: 0049-421-218-64259
fax: 0049-421-218-98-64259
e-mail: christoph.stahl@dfki.de

Acknowledgements to DFKI GmbH and SFB/TR8 Spatial Cognition (project I5-DiaSpace) for the support of this workshop.

Programme

LTPC is a half-day workshop on June 19, 2012.

9.00 – 9.15	Welcome and introduction of the participants
9.15 – 9.45	Keynote: On the role of context sensing for dialogue systems in automotive and AAL environments. ***Christoph Stahl***
9.45 – 10.15	The design of voice controlled assistive technology for people with physical disabilities. ***Mathijs Verstraete****, Jan Derboven, Jort Gemmeke, Peter Karsmakers, Bert Van Den Broeck and Hugo Van Hamme*
10.15 – 10.30	Discussion
10.30 – 11.00	Coffee break
11.00 – 11.30	Multimodal Interaction in Dynamic and Heterogeneous Smart Environments. *Sebastian Bader,* ***Gernot Ruscher*** *and Thomas Kirste*
11.30 – 12.30	Discussion with all participants and wrap up
12.30 – 14.00	Lunch

Objectives of the LTPC Workshop

The main objective of the workshop is to consider the intersection of two research domains which have been separated the past years: Language Technology and Pervasive Computing.

In pervasive and ubiquitous computing scenarios, spoken language is in many cases the ideal modality for human beings to interact with a "disappearing" computer, i.e. to directly formulate their intentions and to receive feedback from the system. However, despite recent advances in speech technology, many developers still have objections to employ natural language processing (NLP) due to concerns of low recognition rates and issues of disambiguation etc. These limitations of NLP components are not surprising, considering that even human beings often can only make sense of spoken language by contextual knowledge. Sensing context, however, is one of the key topics of the Pervasive Computing conference series, particularly location sensing and activity recognition. We believe that the pervasive computing community can make an important contribution to the field of NLP.

Language Technology and Pervasive Computing

Pervasive Computing technology provides many types of sensors that can provide context to a dialogue system.

The location of users within an environment and the surrounding objects are of major importance for dialogue between the user and a system. It is more likely that the user refers to objects that are nearby and visible, so the speech system should consider this for disambiguation. The challenge is to sense and represent the position of the user and other objects in a location model, and to incorporate this model into language processing. Likewise, utterances are likely to refer to the current actions of the user. Much effort has been spent in recognizing and representing activity and NLP components should make use of such contextual information.

Gesture recognition is another pervasive technology that can improve speech-enabled systems through multimodality, with manifold applications. Smart energy scenarios drive the connection of objects with the Internet in order to measure and reduce power consumption. This development offers new chances to remotely control and even interact with objects through speech. Likewise, our ageing society requires technological solutions that allow elderly users to live independently at home (Ambient Assisted Living). Such assistance systems require intuitive user interfaces, i.e. based on speech, to keep the humans in the loop.

The quality of user interfaces may be further increased by personalization through speaker recognition.

Finally, user interfaces of pervasive computing systems must be translated and localized to different languages and cultures in order to be successful on global markets. Hence NLP frameworks should include tool support for efficient and correct translation of resources, such as grammars.

Paper Session

The paper session comprises two papers that highlight the importance of context for dialogue systems. The first paper describes a user-centered design approach, and the second paper is about multimodal interaction in smart environments.

Mathijs Verstraete et al. present the *ALADIN* project, which aims to develop an assistive voice control system for people with physical disabilities. Their position paper describes the user-centered design approach used in the project to identify the users' needs, and to develop the interaction with the assistive technology. To get an understanding of how people address voice-controlled technology, the authors held test sessions with scenario visualizations. One conclusion is that the system should identify the users' intentions based on their location and context (for instance, 'light on' turns on the lights in the room where the user is, without specifying which particular lamp).

Sebastian Bader et al. present the *Helferlein* system, which has been designed and built for the control of dynamic and heterogeneous ensembles of devices and services. The system employs contextual information about the current position of the user, which is analysed and used to identify device instances. Within a smart meeting room scenario, a screen can be selected by stating "On this screen here". Also, the current user situation and activity are considered to disambiguate user utterances. For example, it is usually not appropriate to ask the user during a lecture using the SpeechOutput because this would disturb the lecturer.

Discussion Topics

In the second part of the workshop we will discuss language technology from both the application and vendor perspective. This session aims to work out requirements for NLP, such as speech recognizers or text-to-speech modules in the field of pervasive computing with a special focus on context. Speech technology, such as speech recognition or text-to-speech modules, is commonly used in the pervasive computing community as "black box". However, adding knowledge about the user as well as contextual information, i.e. derived from sensors and ongoing interaction with the environment, could lead to significant improvements of speech and language processing.

One goal is to define recommendations for future research in both fields that address the identified requirements for NLP components:

- identify requirements for NLP from the application engineer's perspective;
- gain insight in language processing systems from the linguist's perspective;
- make recommendations for future research and development of speech technology.

We thank the participants for their contributions and are looking forward to a successful workshop in Newcastle!

Christoph Stahl and Dimitra Anastasiou

The design of voice controlled assistive technology for people with physical disabilities

Mathijs Verstraete[1], Jan Derboven[1], Jort Gemmeke[2], Peter Karsmakers[3], Bert Van Den Broeck[3], Hugo Van hamme[2]

[1]Centre for User Experience Research (CUO), IBBT-KU Leuven Future Health Department, Parkstraat 45 Bus 3605, 3000 Leuven, Belgium
{mathijs.verstraete, jan.derboven}@soc.kuleuven.be
[2]KU Leuven – Dept. ESAT, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
{jort.gemmeke,hugo.vanhamme}@esat.kuleuven.be
[3]Mobilab, KH Kempen (association KU Leuven), Kleinhoefstraat 4, 2440, Geel, Belgium
{peter.karsmakers, bert.van.den.broeck}@khk.be

Abstract. The ALADIN project aims to develop an assistive voice control system for people with physical disabilities. This position paper describes the user-centered design approach used in the project to identify the users needs, and to design the interaction with the assistive technology.

Keywords: Assistive technology, voice interface, user interaction, HCI

1 Introduction

Voice control of the technology that we use in our daily lives is perceived as a luxury, suited only for situations in which hands-free control is appropriate, such as in-car voice control systems. Apart from these specific hands-free situations, more common interactions and input methods are often considered more suitable. A remote control, for instance, can be better suited for home automation because often, it is easier to push a button than to say a command.

However, for people with a physical impairment, pushing a button is not always as easy as it is for most people. For this target group, a wide range of assistive technologies is currently available, including traditional joysticks, touchless finger joysticks, tablets, chin switches, pedals, head-mounted switches, etc. Nevertheless, only a restricted amount of information can be transmitted through these devices. Also the speed of operation and the complexity of the function one wants to accomplish are important boundaries. In addition, the physical effort in using these devices is a burden for some users (see Fig.1).

Fig. 1. Remote controls used by people with a physical impairment.

To increase speed and complexity on the one hand, and reduce effort on the other, voice interaction can be a viable solution. What is perceived as a luxury for most people can actually mean a significant improvement in the quality of life for a disabled person. Furthermore, it has a high social impact for this target group.

The ALADIN project aims to develop an assistive voice control system for people with physical disabilities. This position paper describes the user-centered design approach used in the project to identify the users needs, and to develop the interaction with the assistive technology.

2 Obstacles in Voice-Controlled Assistive Technology

Even though voice interfaces can provide significant improvements for specific target groups, they are currently not widely used for assistive devices for several reasons. Technology for people with disabilities often needs to cope with a high level of variation in user requirements for assistive technology, creating high individual adaptation and development costs. In addition, users for whom voice commands could be of added value, often also have a speech pathology, such that state-of-the-art speech recognizers are unusable for them. Moreover, the user's voice may change over time due to progressive speech impairments.

The ALADIN project proposes an approach that is based on learning and adaptation. The interface should learn what the user means with his/her commands, which words he/she uses and what his/her vocal characteristics are. Users should be able to formulate commands they like, using the words they like and only addressing the functions they are interested in. Learning takes place by using the device, i.e. by mining the vocal commands and the change they provoke to the device. This approach has not been taken by any other commercial available voice system.

3 User-Centered Design Methodology

3.1 General

With regard to interaction design, voice user interfaces (VUIs) are often considered a 'non-traditional' interface. Most interaction design (and subsequently most design methods) is focused on interfaces with a heavy focus on visual information and manual user input, such as software interfaces, websites, physical products, etc. However, in the last decade, more and more research has been carried out into the design of usable VUI's, which has resulted in general guidelines for designing VUI's [1]. Most of this research and design principles are aimed at VUIs for mainstream applications. Research into VUIs for users with disabilities is still scarce.

Both VUI design, as a non-standard interface, and designing for users with disabilities require a thorough Human-Centred Design (or User-Centred Design (UCD)) approach [2,3]. In a human-centered design approach, the end-users are the central focus in the design of new products or applications. This is especially important when the end-users are very different from the designers and developers who might have difficulties empathizing the actual end-users. The problems, needs, tasks and contexts of the end-users are addressed during each phase of the design and development process and end-users are actively involved throughout the entire process. This way, the match between the products or applications under development on the one hand and the user needs on the other hand can be optimized from an early stage onwards. Continuously keeping an eye on the user needs allows for high levels of usability and a positive user experience. In addition, later changes to meet user needs and future redesigns to enhance usability are reduced.

3.2 User and task analysis

In the first phase of the project, we gathered background information of the participants, their pathology, limitations, caregivers and tasks, leisure time, time consumption and living environment. The group of participants ranged within different parameters, grade of independence and limitation. In addition to the interviews, every participant was asked to guide us through their home while telling about and performing their daily tasks, e.g. explaining the tools they use, difficulties they encounter, etc. By doing this we gained a good insight of the abilities and needs of the user.

3.3 Scenario Test Sessions

As the ALADIN system is based on learning what the users mean with a specific voice command, it is important to get a clear view of the variability of the voice related acoustical parameters and type of commands they choose. To get an understanding of how people address voice-controlled technology, test sessions were held with scenario visualizations.

Fig. 2. Scenario visualization.

The scenarios (example in Fig. 2) were presented to the respondents, starting from simple situations to more complex ones. Guided by the scenarios, the moderator asked the respondents to formulate any voice command they wanted, without worrying about any system limitations. The scenario visualizations were used in order to avoid biasing respondents with specific words or sentence structures. In the process, the sketches were completed with additional elements (see Fig. 3), adapting the scenes to the participants emerging system image. For later analysis, the entire process was videotaped.

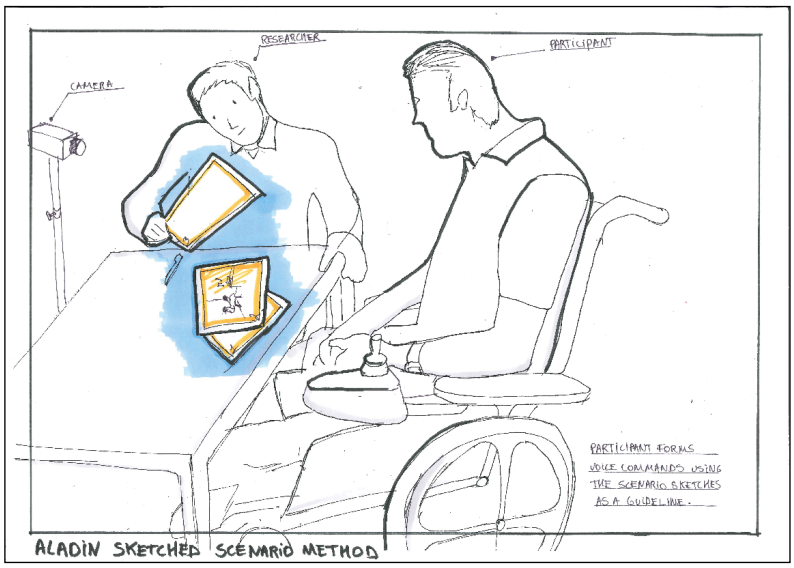

Fig. 3. Sketched scenario method.

The above test sessions provided significant material on how test users address a voice interaction system. Significant variation was found, for instance ranging from a purely 'technical', command-style interaction to a more anthropomorphized, personal communication with the system. This last group tended to make a conversation with the system in a human or natural way, while the command style interaction implies a way of thinking in terms of efficiency and reliability, which is very important for this specific target group. Likewise we distinguished different ways of how people want to communicate and interact with a voice-controlled system. We observed various ways of how people formulate commands; activating and stopping the system, specifying and altering commands. In addition, some respondents addressed individual devices, without addressing the voice-control system separately, while other respondents addressed the voice-controlled system as a whole, telling it to act on the environment and control other devices. For this last group, addressing separate objects such as doors felt very unnatural. On the other hand, the device-oriented way of thinking implies a different technology approach, in which the system can identify the users' intentions based on their location and context (for instance, 'light on' turns on the lights in the room where the user is, without specifying which particular lamp).

4 Functionalities

Based on the user and task analysis we gained insight in the target group and how they organize their everyday life and routines, their tasks and their contexts. Depending on these users' needs and abilities

we will develop different functionalities; home automation, communication tools and entertainment functionalities. For people with limb impairments, successful voice control can have significant impact on the quality of life by facilitating independent living and communication. A voice-controlled environment can contribute to independence of living: being able to perform actions autonomously gives a sense of freedom, increases self-esteem and gives a feeling of regaining power over the environment. For example: Opening a door by your own command instead of asking someone to open a door. Apart from independent living, the system can also extend communicative abilities. A typical application of this technology would be the control over a telephone, which may not be within reach of the user. Relatedly, new ways of calling for help in emergency situations are opened by facilitating communication. In addition to addressing basic needs such as independent living and communication, a voice-controlled system can also provide new opportunities for entertainment. Being able to operate recreational devices such as the television and radio are straightforward – but important – examples. Other avenues that can be explored include gaming, playing chess or solving Sudoku puzzles.

5 Conclusion and Future Work

The ALADIN project aims to develop an assistive voice control system for people with physical disabilities, as voice interfaces can provide significant improvements for this specific target groups. So far, we got insight of how people with a physical impairment differ in communicating with assistive voice systems. In addition to the conventional user and task analysis, we used the sketched scenario method to get an understanding in how people would interact and communicate with such systems without biasing the participants. Future work in the project includes the actual design of the voice user interface, incorporating feedback and correction mechanisms, learning patterns, etc.

Acknowledgements. Organizations involved in the project are KU Leuven department ESAT, KU Leuven CUO, K.H.K Mobilab and the University of Antwerp department CLiPS. This work is sponsored by IWT-SBO project 100049 (ALADIN).

References

1. Cohen, M. H.,Giangola, J. P., & Balogh, J.: Voice user interface design. Addison-Wesley Professional, (2004)
2. Mayhew, D. J.: The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. Morgan Kaufmann, (1999)
3. Sharp, H., Rogers, Y., & Preece, J.: Interaction design: beyond human-computer interaction. Wiley, (2007)

Multimodal Interaction in Dynamic and Heterogeneous Smart Environments

Sebastian Bader, Gernot Ruscher, and Thomas Kirste

MMIS, Computer Science, University of Rostock
firstname.lastname@uni-rostock.de,
http://mmis.informatik.uni-rostock.de

Abstract. Multi-modal interaction is an interesting and challenging research field. It targets at a natural interaction between the user and its environment using different in- and output modalities. In this paper, we present a working system which has been deployed into our laboratory. It has been designed and built for the control of dynamic and heterogeneous ensembles of devices and services. All components of the system, including the supported modalities and controlled devices can change over time. In addition to describing the system, we discuss an example illustrating its usage. While building our system, we tried to keep it as simple as possible while nonetheless enabling multi-modal interaction within dynamically changing environments.

Keywords: Multi-Modal Interaction; Smart Environment; Situation- and Location Awareness

1 Introduction and Motivation

Modern meeting rooms are equipped with numerous devices and provide different services to their users. But with the increasing complexity of such environments, the control and interaction becomes more and more challenging. The multimodal interaction between the environment and its users might help to solve this problem. In particular, we believe that the interpretation of the current context including the state of the devices and the user position is needed and useful to allow the natural interaction with the system.

The contribution of this paper is twofold. On the one hand, we describe how to build a working system for dynamically changing ensembles of devices, users and services. It has been implemented as a modular and decentralised architecture controlled via the contract net protocol. On the other hand, we show how to enable location and situation awareness in such a system. That is, how to incorporate the current user's location as well as the state of the user and the environment into the decision making of the system. The system's architecture has been kept as simple as possible and nonetheless allows multimodal interactions as in more elaborate approaches.

In the following section, we introduce some preliminary notions and give pointers to other relevant work. In Section 3, we describe our system in detail

– the general architecture, available dialogues and input-output modalities. We furthermore discuss use cases to show the system in action and to discuss the context awareness of the system. Finally, we conclude the paper by summarising our work and pointing to future extensions.

2 Preliminaries and Related Work

With this section, we conceptualise our understanding of *Smart Environments* as well as of *Multimodal Interaction*. Based on that, we show which requirements for a Smart Environment middleware we identified during our studies and outline our own approach, the HELFERLEIN system. Finally, *ContractNet* is presented as an interaction protocol which proved appropriate for information retrieval during multimodal dialogues.

2.1 Smart Environments

We assume indoor places to be *Smart Environments*, if they are able to react on the people's activities in a way that provides some proactive assistance (cf. [8] and [6]). In virtually all cases, the kind of environment determines the set of possible activities. In particular, the platform for our studies was our SMARTLAB (cf. [3]), a prototype of a Smart Meeting Room.

Furthermore, people may at any time bring in personal devices (PDAs, smart phones, laptops, and the like), which should then integrate with the existing devices in a spontaneous and seamless manner. To make things even more complicated, almost all devices are manufactured by different vendors and implement different protocols. Nevertheless, brought-in as well as preexisting devices should build up a heterogeneous, dynamic and ad-hoc device ensemble to assist the user. Given those abilities, even a "white room" scenario offering no pre-existing devices or infrastructure, should not present a problem, and building up a Smart Environment solely from brought-in devices should require no engineering skills on the part of the users. During the remainder of this paper, we assume Smart Environments to have those capabilities.

2.2 Multimodal Interaction

System architectures for multimodal interaction using multi-agent environments have been investigated for quite a while. Some examples are the Galaxy Communicator [4], the Open Agent Architecture [7], the EMBASSI model [10], or, more recently, the Context Aware Multimodal Interaction Model [11].

According to [5], modalities are communication channels, offering different ways of interaction between the system and the users. Typical modalities are graphical user interfaces (GUIs) and natural speech as well as gestures and mimics. Beyond that, we assume sensor data as additional modalities. E.g. location information can help to disambiguate between location-specific devices or services. While the former modalities are potentially bidirectional, sensor data

brings information from user to system only. Please note that this kind of location awareness is related to locations within a Smart Environment and does not include anything like e.g. country-specific language selection. We further assume the speech modality always to be monolingual.

A dialogue is viewed as a (possibly branched) sequence of interaction steps between user and system. Hereby, the system tries to obtain missing parameters in order to fire assistive device or service actions. Such an information retrieval process may be initiated by some specific event, e.g. a speech input like "Switch off the lamp!" or "Which slide is currently shown?".

An emerging problem is the fact that at any point in a dialogue any modality may be chosen by the user. E.g. when a selection menu has been presented by a display, natural speech should also be applicable to select the desired item. Furthermore, dialogue-initiating events may occur at any point in time, even when the current dialogue has not yet been completed. And finally, which modality should the system use in which situation? In contrast to other elaborate approaches, the objective of our work has been to identify a system concept that is as simple as possible for achieving context-aware multimodal interaction for dynamic device ensembles.

2.3 Middleware Requirements and the Helferlein System

Because of the above-mentioned ad-hoc systems dynamics of Smart Environments, it is clear that when we build such a system, we cannot rely on any static device or service information, but need to discover them in an ad-hoc manner through a *Look-up Service*, which enables us to specify our requirements and have a matching device or service located.

Furthermore, devices entering an ensemble need at any point in time ways to get all necessary information, i.e. the required subset of the entire world state, at a glance. Our approach to this requirement is a *Tuple Space* as central information service, a shared associative memory which can store information in the form of tuples: All stored information can be queried using templates. One or all stored tuples matching such template are returned. This brings about a decoupling in space and time: Devices and services do not need to know where others are located or when they are ready to communicate.

Once enough information has been retrieved, and device actions are ready to be fired, the need for a *Remote Procedure Call* (RPC) mechanism comes up. And finally, in order to stay informed about world state changes, an eventing system is required, possibly as *Publish-Subscribe* system.

The mentioned requirements led us to the development of HELFERLEIN [2, 3], a middleware specifically tailored to build dynamic heterogeneous ensembles – designed for easy prototyping and usage in research and teaching. The underlying idea is that of a distributed set of objects, deployed into some network, while providing means to interact with those objects using different channels.

2.4 Contract Net

The FIPA *Contract Net Interaction Protocol* [1] is an agent interaction standard which gives an agent (the *initiator*) the role of a manager wishing to have some task performed and at the same time to optimise a function characterising the task, possibly as its cost (e.g. soonest time to completion). For that to happen, the initiator spreads a *call for proposals* (cfp), which contains the task itself, the related conditions as well as a deadline. Other agents (the *participants*) may either respond with a binding *proposal*, comprising the agent's conditions, or refuse to propose. After the deadline has passed, the initiator may accept or reject one, several or all proposals. Once having performed the task, accepted participants (the *contractors*) return with the result or a failure report.

Within our implementation, a *ContractNetManager* component encapsulates the initiator's protocol acts, i.e. spreading a cfp, accepting and rejecting proposals as well as gathering the results. For this purpose, the ContractNetManager offers procedures to obtain the results of either all proposing participants or a single one, whereas the selection is based on a given policy. Furthermore, the ContractNetManager offers broker functionalities to select and return appropriate participants.

3 Multimodal Interaction in Dynamic Environments

In this section we introduce and describe the system implemented and deployed into our laboratory. It has been designed to enable multi-modal interaction within dynamic ad-hoc ensembles of heterogeneous devices. That is, the system has been built such that the controlled devices, the available modalities and the available dialogues can change over time. The only required component is the system's *dialogue manager* – a component controlling the available resources, invoking dialogues and controlling the progress of a running dialogue.

The in- and output is realised using *interaction device*s, described below. Those devices encapsulate different modalities to interact with the user. The sequence of interaction steps is described and implemented within *dialogue*s. At run-time, the currently available devices are identified and invoked using the Contract Net protocol described above.

3.1 Available Dialogues

A dialogue can be understood as a sequence of user interactions and device actions, choices, loops and sequences. In our system, dialogues are realised as Java components implementing a common interface. Note that these preconfigured dialogue programs are conceptually quite similar to the hub scripts outlined in [4].

As mentioned above, all dialogues can enter and leave the ensemble at any time. In particular, we do not presume the availability of any of the dialogues described below. The middleware registers all currently available dialogues within a look-up service and within the contract net.

Whenever the dialogue manager receives an interaction event, emitted by an interaction component described below, it identifies matching dialogues via contract net. For this, a call for proposals is spread within the ensemble and every matching dialogue answers with a proposal containing a rating – that is an information describing how well the dialogue fits to the interaction within the current situation. The best matching dialogue is chosen and executed.

Table 1 shows a list of pre-defined application independent dialogues available. For example there are dialogues for debugging and maintenance tasks. In addition, application dependent dialogues can be added as described below.

EchoDialogue repeats the last speech utterance with the remark that it could not be interpreted by the system
DialogueHelpDialogue reacts on the command "show dialogue help" and lists all available dialogues together with a short description
ShowDevicesDialogue reacts on the command "show devices" by showing a list of all available devices together with their available methods
DeviceExecuteDialogue reacts on (partial) device execution commands like "dim to 0.5" by (1) completing the command by asking the user for missing information (e.g., which lamp should be dimmed) and (2) executing the command by invoking the corresponding method on the specified device

Table 1. Predefined and application independent dialogues being part of the system.

3.2 Interaction Components

To allow the interaction with the user, different interaction components have been realised. Interaction component are realised in Java by implementing a given interface. As for the dialogues, the best matching component is identified via contract net. After discussing some of the realised components we describe user and system initiated interactions.

Available Interaction Components The following components are available within our system Different *Graphical User Interface (GUI)* have been designed to allow the presentation of messages, choices and arbitrary inputs, for device selection, device-method selection and to display tabular output to the user. To allow the usage of *speech inputs* we use a standard Windows 7 PC with the built-in speech processing capabilities. Whenever the system recognises a user utterance, an input event is distributed. To generate *speech output*, we use the language generation feature included within the MacOS X system. As the current *user position* plays an essential role for the disambiguation of the user's utterances, we encapsulated our indoor localisation system within an input modality. It is used to answer questions relating to the position of the user. For this, we analyse the raw and noise sensory inputs using a hidden markov

model and convert it to symbolic positions like *near screen1* etc. To allow a very natural interaction between user and environment, we also included other sensors. In particular, we use our self-made VGASensor to detect whether a laptop's video output has been connected to lab. A PenSensor is used to detect the usage of a whiteboard.

User Initiated Interaction We assume, that the user may start an interaction at any time. Therefore, all available input components have been designed to broadcast events as soon as a user interaction has been noticed.

System Initiated Interaction Whenever a user interaction is necessary, a call for proposals is spread which contains a specification of the required interaction. For this, we rely on interaction performatives as known from KQML [9]:

- Ask. Present a question to the user and wait for an answer.
- Choice. Equivalent to ASK, but with a limited set of allowed answers. Can be shown as a set of buttons with a GUI, or by reading the list to the user and asking for his choice.
- Verify. Verifies a given statement by presenting it to the user together with an option to accept or reject it. Can be done via GUI or speech interaction.

Within the example described below, a device instance needs to be selected. For this the following call for proposals is distributed among the contract net participants: `(ASK :what device-id :device-type screen)`.

3.3 The System in Action

Below we discuss a small example to show our system in action. All devices currently present within our laboratory register themselves within our middleware. That is, they announce their presence, current state and capabilities. In particular every device provides a list of supported methods which can be invoked by the user. This information is stored within a tuple space and state changes are distributed using our Publish-Subscribe system discussed above.

As soon as the user connects a laptop using a VGA cable, the VGASensor mentioned above, recognises this and distributes a sensor-interaction event. The resulting Contract Net negotiation selects a dialogue capable of reacting to the event and executes it. In our lab, an application dependent dialogue is used for this. It will query the user for a screen to be used for the projection by spreading a Ask query. Assuming the user is close to a monitor, the question is displayed using the GUI. If the user does not answer within a given time interval, the question is repeated using a different modality as for example the SpeechOutput. After presenting the question to the user, the dialogue waits for an answer. The user can answer the question by moving to a screen and stating "*On this screen here*". After resolving the "*here*" using the user's current position, the system executes the necessary device actions to show the presentation on the specified screen.

Multimodal Interaction The example above shows the multi-modality implemented within the system. A sensory input is used to trigger a dialogue (modality 1). The system asks the user by emitting GUI (modality 2) and SpeechOutput (modality 3) interactions. The question is answered by SpeechInput (modality 4) and UserPosition (modality 5). To allow an ad-hoc selection of the most suitable modality, the dialogues trigger all user interaction using Contract Net instead of assuming the presence of one particular modality and directly linking to it.

Location Awareness As described above, the system is able to employ position information. The current position of the user is analysed and used to identify device instances. In particular, the system is able to find the device closest to the user providing a specified functionality. Within the example above, a screen needs to be selected and the user answers the question by stating "*On this screen here*". The "*here*" together with the knowledge that a screen is needed, allows the system to select the one which is closest to the user.

Situation and Activity Awareness As mentioned in the introduction, the current user situation and activity can help to disambiguate user utterances. For example it is usually not appropriate to ask the user during a lecture using the SpeechOutput because this would disturb the lecturer. Therefore, the current state of the environment and the user actions should be taken into account. In our system, such situation dependent information is stored within the tuple space and can be used by every component on demand.

4 Conclusions and Future Work

In this paper, we discussed how to enable multi-modal interaction within a smart environment. The system has been designed to allow a natural interaction between the user and a distributed, dynamic and heterogeneous ensemble of devices and services. It has been implemented and deployed into our Smart Appliance Laboratory – a real hardware system.

To separate the flow of the interaction and the concrete modality to be used, the functionality has been implemented within different components: *dialogues* controlling the sequence of interactions and *interaction components* encapsulating one particular modality each. The connection between the different components is established via the Contract Net protocol. This allows to adjust the system to the currently available modalities, devices and dialogues ad-hoc and without a new setup phase.

While evaluating our system, we found the interaction with the environment to be very natural, even though far from being perfect yet. The current implementation of the speech input needs to be improved. To guarantee a robust recognition of devices and methods, no synonyms are included into the grammar. Therefore, the user has to use the correct words to refer to those entities.

Another important aspect, not addressed within our system yet, is the multi-user interaction with the system. In particular while resolving position dependent information, we assume the presence of one person only – to be more precise, we assume that there is position information of one person only. Even though our localisation system is able to track multiple users simultaneously, this information is not yet integrated into our multi-modal interaction system.

Furthermore, the available implementation of the system does not yet utilise very sophisticated decision algorithms. Currently, simply the highest-rated action becomes fired. This behaviour is not optimal in situations, where likelihoods have only little difference, which means in fact, that the system is not "sure" at all. But the integration of probabilistic reasoning techniques should not be a problem in principle for the presented system.

Acknowledgements Sebastian Bader's work is supported by the DFG within the graduate school MuSAMA (GRK 1424). Gernot Ruscher's work in the MAike project was supported by *Wirtschaftsministerium M-V* with means from *EFRE* and *ESF*. The authors wish to thank their students Christian Eichner, Nils Faupel, Stefan Gladisch, Daniel Moos und Martin Nyolt, who implemented the system.

References

1. Fipa contract net interaction protocol specification (2002)
2. helferlein: The helferlein System, a Middleware for Dynamic and Heterogeneous Ensembles (MAR 2012), https://code.google.com/p/helferlein/
3. Bader, S., Ruscher, G., Kirste, T.: A middleware for rapid prototyping smart environments: Experiences in research and teaching. In: Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing. pp. 355–356. ACM, Copenhagen, DK (SEP 2010)
4. Bayer, S., Doran, C., George, B.: Exploring speech-enabled dialogue with the galaxy communicator infrastructure. In: HLT '01: Proceedings of the first international conference on Human language technology research. pp. 1–3. Association for Computational Linguistics, Morristown, NJ, USA (2001)
5. Bui, T.H.: Multimodal dialogue management—state of the art. Tech. Rep. TR-CTIT-06-01, Centre for Telematics and Information Technology, University of Twente, Enschede (2006), http://eprints.eemcs.utwente.nl/5708/
6. Chen, L., Nugent, C., Biswas, J., Hoey, J. (eds.): Activity Recognition in Pervasive Intelligent Environment. World Scientific, Paris, France (APR 2011)
7. Cheyer, A., Martin, D.: The open agent architecture 4(1–2), 143–148 (2001)
8. Cook, D., Das, S.: Smart Environments. Wiley (2005)
9. Finin, T., Labrou, Y., Mayfield, J.: Software agents, chap. KQML as an agent communication language, pp. 291–316. MIT Press, Cambridge, MA, USA (1997)
10. Heider, T., Kirste, T.: Architecture considerations for interoperable multi-modal assistant systems. In: Proceedings of the 9th International Workshop on Interactive Systems. Design, Specification, and Verification. pp. 253–268. Springer-Verlag, London, UK (June 2002)
11. Luo, Q., Zhou, J., Wang, F., Shen, L.: Context aware multimodal interaction model in standard natural classroom. In: Proceedings of ICHL'09. pp. 13–23 (2009)