

Implicit Interaction for Pro-active Assistance in a Context-Adaptive Warehouse Application

Oliver
Kaufmann
+49 2241 14
2207

Andreas
Lorenz
+49 2241 14
2204

Reinhard
Oppermann
+49 2241 14
2703

Alex
Schneider
+49 2241 14
2869

Markus
Eisenhauer
+49 2241 14
2859

Andreas
Zimmermann
+49 2241 14
28561

Fraunhofer FIT
Schloss Birlinghoven
D-53754 St. Augustin

{Oliver.Kaufmann}{Andreas.Lorenz}{Reinhard.Oppermann}{Alex.Schneider}
{Markus.Eisenhauer}{Andreas.Zimmermann}@fit.fraunhofer.de

ABSTRACT

This paper reports on the implicit interaction evaluation for pro-active assistance support of warehousemen developed in the MICA (Multimodal Interaction in Context-Adaptive systems) project. The goal is to evaluate natural human interaction in multimodal systems and the provision of pro-active assistance: we aim to do this by combining explicit and implicit interaction on different modalities reflected in a new layer based architecture for multimodal interaction.

Keywords

Context Awareness; Multimodal Interaction; Pro-active assistance; Layer based Architecture

1. INTRODUCTION

MICA (Multimodal Interaction in Context-Adaptive systems) aims at implicit human interaction with multimodal systems. To realize this vision we try to evaluate the natural behavior of human beings as implicit means to control the information system and to receive pro-active system support in situations the user needs help. Explicit interaction of the human with an information system focuses on the interaction with specific IT-devices (screen, tablet, keyboard, mouse) defined by the component designer and focuses on interaction methods (type, click, drag) defined by the rules of the interaction designer. Implicit interaction means that the user's behavior focuses on the arranged or natural environment by movements of the human in the physical space and is at the same time indicative as an interaction with the IT-system. For understanding implicit interactions an interpretation is needed about the task and intentions of the user in the specific context. There are a lot of applications that can profit from implicit interactions of the human because less training and less fixation of the

human to the IT-system or the IT-interface components is needed. It is not necessary to talk about a "user" of an IT-system but we can talk about a "human" who interacts with the complete environment and the IT-system is part of the environment in the sense of ubiquitous computing, ambient computing, or disappearing computing.

Especially in applications for which extensive user training is not feasible, applications need to anticipate a mix of purposive behavior and uncertainty, to render information in a way that is compatible with the characteristics of the terminal, the situational context, the cognitive load, and the user's personal preferences. In these situations multimodal interaction can be effective. Multimodal interaction comprises several interaction modes like typing, clicking, speaking, hearing and can include implicit interaction like gesticulating, grasping, watching, walking. To refer to multimodal interaction and not limit the interaction to implicit interaction should contribute to the reliability of the interaction. Implicit interaction is comfortable but it is also sometimes obtrusive and vague.

We implemented a system to assist warehousemen in the picking process. The warehouse scenario poses a real challenge for multimodal interaction. Because the workers are always using their hands, hands-free support is required. The environment is often very noisy and the light conditions might change as well. The interaction needs to use different modalities according to the needs of the current situation and task. Warehousemen often have to work under time pressure requiring a very responsive system. Intelligent fusion of fine grained tracking with RFID technology, in combination with pen or speech input and adaptable audio and graphical output will lead to natural blended interaction with the MICA-system.

The disadvantage of most approaches to multimodal interaction especially when gesture input is concerned is that the action to be performed by the users is often socially obtrusive. Excessive gestures performed without a human counterpart look strange to a casual bystander. Our approach is to take advantage of natural behavior: It's a very natural thing, that if you are interested in something you approach the object to get a closer look and to investigate it in more depth. The movement of a worker is also used to determine if he is in need of help and in which modality

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

pro-active assistance has to be delivered. E.g. in a warehouse there are quite noisy areas and in other areas outside of buildings bright sunlight might prevent that a screen is readable (see Figure 1). Obviously a corresponding change of modality is of great benefit.



Figure 1: Situations in areas of the warehouse, where noise or light conditions request a change in modality

MICA demonstrates the advantage of sensor fusion for a new interaction paradigm through movement of the users in the environment. The paradigm is portable to diverse application scenarios; possible application scenarios range from maintenance, warehouse, training activity support to IT-management support.

2. CONTEXT-AWARE and MULTIMODAL SYSTEMS

Context-aware computing is a computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people, physical environment and technical infrastructure). The relationship of these components of the context is constantly changing due to user activities or external processes. In an early approach to model the context of a mobile user, Schilit et al. [1] present a facility for mobile application customization called “dynamic environment variables”. Since it was proposed about a decade ago, many researchers have studied this topic and built several context-aware applications to demonstrate the usefulness of this new technology.

From a general point of view, Dey and Abowd [2] define context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an applica-

tion”. Focusing on what kind of information describes the subject’s context, we condense the definitions of Schilit et al. [3] and the four context-dimensions of Gross and Specht [4]: User context, computing context, time context and environment. Additionally, when the context-information is recorded across a time span, we can obtain histories like interaction history, movement history, or event histories, which could also be useful for certain applications [5]. Context awareness includes the history of the context components. Based on this context history for each situation the specific constellation of the context determinants have to be evaluated to infer the best guess for the intention of the user. Situation awareness is thus a snapshot of context awareness for a given time and space [6]

Currently, context-awareness lacks standards for exchanging information as well as a generic architecture and processing pipeline. Through the course of the MICA project, we developed an architecture hosting several components to achieve reusability, domain-independency and efficiency in the developing process. Most important we aimed at mapping the contextualization process to corresponding layer based architecture. The contextualization process is divided into four steps, each corresponding to one layer in the architecture: Sensoring – recognizing the relevant parameters, Modeling – analyzing information and assigning meaning, Controlling – dialogue management, and Rendering – Generating appropriate output.

Recent architectures for multimodal applications (e.g. [7, 8]) have especially achieved success in support for sensing the multimodal user interaction, recognition of valuable input (like [9]) and disambiguation of recognition errors [10]. This research results in high-end abilities for recognition and synthesis in common modalities such as speech and handwriting.

Pro-active assistance is an area of research which focuses on assisting people during work. Common approaches to infer the user need for help by observing the actions of users and inferring their goals with these actions by using simple rule-based systems like finite state machines or more sophisticated probability based systems like Markov Chains [11] and Bayes’ theorem [12] which already have proofed good suitability.

3. SYSTEM ARCHITECTURE

For the system to identify situations described in the last chapter where pro-active assistance is needed a client server architecture was developed that evaluates several types of input about the current situation including implicit interaction based on sensors with the user and in the environment (see Figure 2).

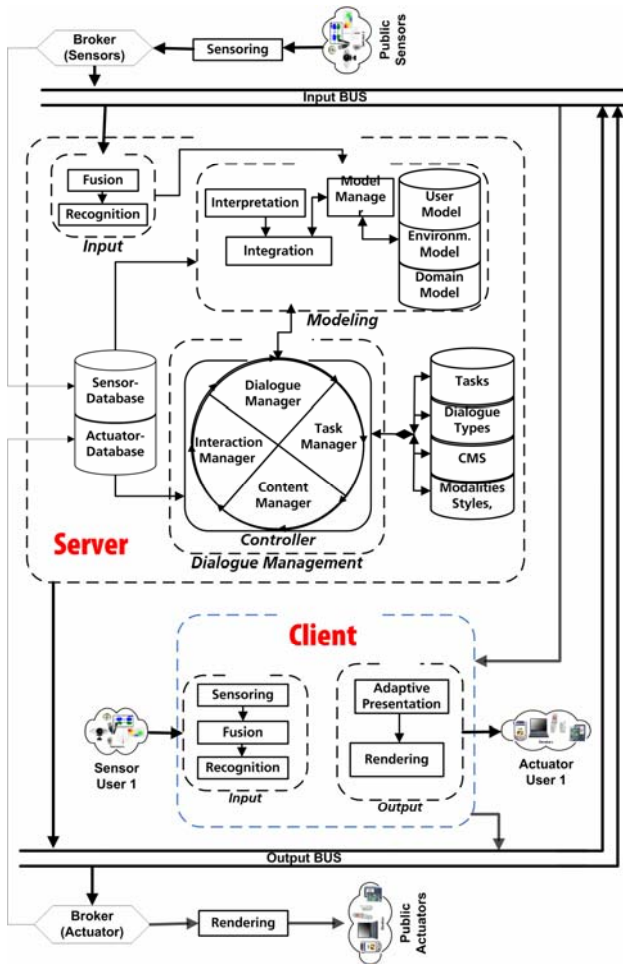


Figure 2: MICA Architecture with centralized and distributed processing

In our work, we go beyond observing input from the different modalities to also integrate the recognition and interpretation of meaningful user-related and environmental parameters. From our point of view, the crucial tasks of building and maintaining a meaningful user-model as well as an environmental model are currently underrepresented. For structuring our components we are also aware of the three layers of the Seeheim model for UIMS design [13]:

- The top layer contains the input and presentation layer
- The middle layer is the dialogue management
- The bottom layer is the application knowledge base

As visible for example in [14], this approach drives the middle-layer to host almost all components (here experts performing specific tasks). In combination with our own research work in context-aware systems [15] and the structure proposed in [16], we defined the MICA architecture, which will be described in detail in the next chapter.

4. THE MICA ARCHITECTURE

It is essential to improve our understanding of the interactive capabilities that are most important for an automated system to conduct a natural multimodal dialogue. We need to integrate a wide range of behavioral data from human users interacting with multimodal applications. Such analysis provides concrete hypotheses for directions to improve components of open multimodal context-adaptive systems, as well as the overall design and architecture of such systems.

The MICA architecture and its contextualization process is divided into four steps: Input, Modeling, Dialogue Management and Output, fulfilling each a specific role corresponding to a single layer in the architecture but residing and being implemented in a mix of centralized and distributed processing, with central processes and layers - Modeling and Dialogue Management at the Server side and Input and Output on the client side.

4.1 Input – Processing Incoming Raw Data

The input layer consists of sensing and recognizing components. Sensors of a multimodal and context-sensitive system perform two tasks: recognition of the user's interaction with the system and sensing of user-related (e.g. user tracking data, head position, user's heart beat) or environmental contextual parameters (e.g. room temperature, light conditions). An interactive multimodal implementation will use multiple input modes such as audio, speech, handwriting, keyboard, and other input modes. Additionally, several context-parameters need to be observed to enable the application to adapt its behavior and to be able to interpret the input of the user correctly. Therefore, a set of physical sensors observe variable parameters of a domain, measure their current values, and make the data stream available for recognition components. Examples for sensors are:

- Microphone, pen, pointer, and keyboard for multimodal interaction.
- Tracking-sensors, (head-)orientation sensors, light and noise sensors, or clocks for contextual information.
- RFID-readers, scale that deliver other useful task related data.

The result of the sensing process is raw digital data not yet specific for any application. The data is forwarded to recognition components which capture natural input from the user and translates it into a form useful for later processing. Multimodal recognition extracts commands from speech or gestures and patterns of implicit feedback, e.g. the recognition of movement patterns. As a result, recognition delivers what input the user provided to the application, for example

- Converting spoken speech, key presses, or handwritten symbols into text.
- Converting pointing gestures into direction and angle, or button presses into (x,y)-positions on a two-dimensional surface.
- Converting different satellite-positions into GPS-coordinates.

Treated as a special enhancement of sensors, a fusion step was placed in-between the sensing and recognition step. Regarding context parameters sensor fusion means aggregating the results of

different sensors together to produce a reasonable approximation of the system state. This means that even if some sensors fail or give erroneous answers, the system will still be able to determine the current state. Different statistical methods might be suitable like taking the median instead of the average, or defining a critical value e.g. all values above the standard deviation are ignored. This method would avoid drastic measures being taken by the context system to correct what it considers to be temperature variations but are actually simply the result of sensor malfunctions. The result is robust system behavior.

Besides other sensor-related data, the sensor database contains grammars, vocabularies, templates and patterns necessary for a successful recognition process.

4.2 Modeling – Information Interpretation and Integration

On the modeling layer, interpretation and integration components analyze the data and put the information into relation with the information about the user and the context. The interpreting step answers the question, what the current value of a sensor means to a specific application. Each interpretation component identifies the "meaning" or "semantics" intended by the user and adds that to the recognition. The emerging knowledge is placed in different models to reflect aspects of the user and user-related data, as well as to describe the environmental context. Examples for interpretation are:

- Interpretation of recognized spoken text to derive its meaning (e.g. is it a command, and which command is it? - for example "display here").
- Interrelation of positioning system data with the spatial model and user model to derive the absolute location (i.e. the room) of the specific user.
- Interpretation of the direction and angle of a pointing gesture to derive the targeted position, e.g. something left to the user.
- Interpretation of the user motion (e.g. is she moving slow or fast?).

The integration step combines output of several interpretation components. Intelligent modules aggregate information by interpreting the data with meta-information. In excess of multimodal fusion, integration puts the meaning of several interpreted information sources into relation, like

- Combining the positioning data from a user with the distance measures to get the distance and relative location of the user to other entities (e.g. "in front of", "far away from", "in eyesight of").
- By integration of the user's position, orientation, and her spoken command ("display here"), and the direction and angle of her pointing gesture towards the left wall, the application determines the user's demand to use this wall as display.

The integration of the incoming information with existing knowledge about the user and her context and other multimodal input refines the knowledge-base and can be used for disambiguation. The environment and user model can be used to resolve ambiguities by considering the contextual parameters defining e.g. the co-

occurring situation and task. The different modes of a multimodal system are not simple analogues of one another and don't involve redundant but complementary information and can be used for mutual disambiguation [10]. The ambiguity in one modality can be resolved by the input of the other.

4.3 Dialogue Management – Controlling content selection and presentation

The dialogue management layer plans, assembles and refines sequences of commands to control the behavior of the system. At this stage, the decisions to be taken by the system include issues like privacy or pedagogical objectives, and handle the dialogue with a set of users in cases where different users are to be reached by the same or different output channels.

In this layer, the content manager selects the appropriate content from the content management system and the interaction manager selects the appropriate modality. The user model particularly determines the selection of content to support the user by providing the right information. The information about the user's environment will be taken into account as well in order to adapt the presentation to the right connotation and intonation.

In order to distinguish between the different parts of dialogue management and to define the processes more precisely, we divided the dialogue management package into three parts, which actually interplay with each other in a parallel process. A dialogue manager, an interaction manager and a content manager jointly co-operate, in order to take appropriate actions, that base on derived knowledge. The dialogue manager as such is responsible for the coordination of the content manager whereas the interaction manager queries them and combines their suggestions.

The content manager is responsible for managing the information-flow of the content-repository. From the research work in context-aware systems of Zimmermann and Lorenz ([15, 17]) we concluded that it is not enough just to supply content without the consideration of the recipient, her current task, and situation (the time, the location, the particular technical environment, and even biological data). Context-sensitive content and information processing is an asset for the generation of added value.

The interaction manager creates and traces certain interaction strategies (e.g. in case of an emergency, it presents information concisely, or uses examples extensively in a learning situation). Additionally, it adapts to certain conditions in the environment or in the user context (e.g. to switch from audio to visual output, if background noise rises). The adaptive or strategic methods are implemented domain-independently, although the expressivity of a modality might be domain-dependent.

4.4 Output – Adapt and Render the Presentation

The output layer handles the connection back to the users. The presentation of the content is adapted to the different technical capabilities of the targeted devices (e.g. screen size). The processed content is then rendered to the appropriate output channel provided by the device. Thus, decisions that have been taken by

the content and interaction manager in the preceding layer result in real world actions.

For configuration and properties of the output components we also added an actuator database with the required information of connected actuators.

5. PRO-ACTIVE ASSISTANCE

A goal of our research in the MICA project was to implement a system that offers best possible support to warehousemen during the process of picking goods. Manual picking is the most error prone process in the whole workflow of finalizing orders in a warehouse. Warehousemen have to work under time pressure and the costs resulting from delivering wrong goods to customers can be very high. This causes a permanent stress to warehousemen especially to inexperienced workers. A pro-active system that could help those workers by observing them during their work and giving unobtrusive and adequate help when they are about to make errors or seem to be in need of help could reduce this stress.

In a field study in two non-automated warehouses we conducted several interviews with the warehouse workers and the warehouse managers and most important we accompanied and observed the workers during their work with incoming and outgoing goods and picking. The aim was to get to know the authentic processes in a warehouse in order to be able to identify situations where help is needed and non-intrusive assistance could improve the work.

Picking of goods means collecting all items contained in an order list in the correct amount. It basically consists out of two steps:

- Getting from the current location within the warehouse to the storage location of the next item to be picked
- Picking the right amount of goods

Since this process is in principle straight forward with a clear goal we didn't have to spend much effort in detecting what is the current task of the worker in order to be able to provide adequate help. The task identification can be based on the electronic order list chosen by the worker. Once selected the goods can be put into a meaningful order minimizing the distances within the warehouse and taking care that fragile goods are not at risk. Though we observed that more experienced workers normally picked several orders simultaneously, which increases the potential list of current tasks to be monitored by the system.

In the warehouse we focused our observation on common errors and time consuming situations that can occur during the picking process. We identified several typical situations in which offering help to the warehousemen could reduce errors and save time. A good example for the situations we identified is the following:

A warehouseman is on his way to the storage location of the next item he has to pick. When he gets closer to the storage location he starts to walk through the shelf rows he passes by and examines them one after the other. "Getting closer" and "examining them" are examples for explicit interaction in the physical space and implicit interaction for the pro-active assistance system.

This behavior, "getting closer" and "examining them", indicates that the warehouseman doesn't seem to know where exactly to find the item he has to pick, that's why he starts a sometimes very time consuming search. It is obvious that offering help in this

situation, telling the warehouseman the exact position of the item would be very helpful.

Another common situation that we could observe in the warehouse was an item in the picking list that was missing or broken. In such a situation the warehouse manager had to be contacted. This initialized a very time consuming process. The warehouseman had to go back – often from the far end of the warehouse to the back office of the warehouse manager. Then the warehouse manager returned with the worker to inspect the place and the missing or broken item, returned again to the office and entered the new information into the system. During that time the process for the worker was on hold.

Now that we gathered some situations in which offering help proactively to the workers should be useful we tried to find a way to model those situations

5.1 Context modeling for pro-active assistance

A simple but very effective solution to alleviate the last described situation with broken or missing items was to provide the worker with a small video camera mounted on the hardhat of the worker (see Figure 3). The warehouseman had the possibility to establish a short video conference with the warehouse manager. This camera could be turned on on request providing the warehouse manager the possibility to look over the shoulder of the warehouseman. This simple solution proved to be quite effective and time saving solution to the procedure described above.



Figure 3: Warehouseman equipped with small video camera

As already stated above our aim was to provide unobtrusive pro-active assistance to the warehousemen while picking or processing goods in the incoming or outgoing area. Our observation and analysis of the processes in the warehouses led us to the conclusion that there was no need in implementing sophisticated probability based systems like Markov Chains and Bayes' theorems. So we attempted to realize a rule-based assistance.

The identification of tasks and goals of the user can be deduced from the list of orders and items to be processed. So we chose an approach which only focused on modeling and detecting behavior identified during our observations of the picking process indicating that the user is in need of help. The detailed analysis of the behavior revealed that we could break the behavior up into eight basic steps out of which each situation can be deduced. These are:

- a) The position of the worker does not change for a certain period of time.

- b) Worker takes the wrong direction (with respect to the storage location of the item he is supposed to pick) for a longer time now.
- c) Worker took the wrong direction for a short time (with respect to storage location of the item he is supposed to pick) but returned to the correct path again.
- d) Worker has picked a wrong item.
- e) Worker is pacing within the warehouse.
- f) Worker starts interacting with his portable device.
- g) Worker picks the correct item.
- h) Worker is looking about/moving his head horizontally in a noticeable way (scanning for a specific item).

A simple straightforward and easy way how to model all those situations within a model is to use a finite state machine. A finite state machine offered us an easy way to reproduce all these situations. The starting state of this state machine is represented by the start of picking and the current pick of the correct item. The final state of the resulting finite state machine marks the point where our system offers help to the warehouseman. A change of the current state of the state machine is triggered by an occurrence of one of the basic steps we defined above. The finite state machine that we actually used for our implementation is shown in Figure 4.

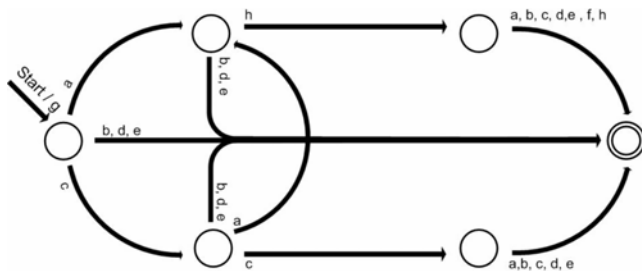


Figure 4: Finite state machine modeling situations in which help can be useful

The process of building this finite state machine also made clear what other contextual information we had to gather so that we could say for example step a) of the list of basic steps mentioned above -position of worker didn't change- has occurred because it is an indicator that the worker needs help and not because he is just talking to one of his colleagues. Navigation in the physical space has to be evaluated together with other context parameters like in the given example the context parameter of social interaction (talking with a colleague).

Though we knew that a finite state machine is a rather primitive way of modeling these situations the later evaluation of our system proofed that it was adequate enough for our approach.

5.2 Detecting the basic steps

Now that we knew what actions of the warehouseman we wanted to observe we had to find ways how we could actually identify and detect these actions.

We used an indoor tracking system to get the exact position as X, Y coordinates of the warehouseman and for building up a model of where to find the different items in the warehouse according to those coordinates. With this data we were able to calculate the

shortest way from the current location of the worker to the item he has to pick up and we could also evaluate if the worker was following this way correctly. This offered us enough information to detect steps a), b), c), and e).

In order to detect steps d) and g) we supplied the trolley of the warehouseman, which they use to carry the goods in the warehouse, with RFID readers and assumed that every item in the warehouse is tagged with a RFID tag.

We were able to detect step f) directly by observing input events on the portable device that we attached to the trolley to display the list of items the warehouseman has to pick. We also used the display of this portable device to provide the actual help to the warehouseman.

To detect step h) we used the same small video camera attached to the hardhat of the worker, that we used to enable the warehouse manager to look over the shoulder of the warehouseman. We implemented software that analyzes the video stream of this camera and that is able to detect noticeable horizontal movements in the single frames of the video stream.

Based on the detection of all eight steps we needed to reassemble the situations we implemented a prototype to detect these situations.

5.3 Validation of the prototype

We validated our prototype by simulating the warehouse scenario. We asked 15 people to take part in this evaluation. We first showed them the small warehouse we build up in our department and asked them to pick a small order consisting of two items. In order to really get the subject in situations where they would need help to find the items and finish their work in a reasonable time we prearranged the items in our shelves in random order. There were as well items in positions on the shelf that were partially hidden by other objects simulating real situations that we observed during our observations in the warehouses where several items were hard to find because they were covered by other items. We tried to verify three objectives in this validation:

- Do the situations in the model correspond to real work-day situations?
- Is the provided assistance helpful and acceptable?
- Does our prototype detect these situations correctly

The validation verified that the situations we modelled represent appropriate events for providing assistance to warehousemen. Three quarters of the subjects really showed the expected behaviour modelled in our finite state machine during the validation. Our prototype was able to detect more than three quarters of the potential assistance events. In these events we asked the subjects if they would appreciate help in their current situation. Except one person all subjects declared that they would be glad to receive help in those situations and to be able to find the items faster.

The results of this validation revealed that our simple finite state machine successfully identified the diagnostic situations and the provided pro-active assistance was of great help for warehousemen and helped to make picking more effective, no item was

missed, and efficient, the pro-active help saved time in finding the items.

6. CONCLUSION

Our approach in MICA was to enhance current multimodal applications to consider more than speech and gestures and to define architecture for flexible, adaptable and pro-active systems. From our research in context-aware system development we concluded to integrate other non-intrusive modalities to support natural blended interaction. In particular, the movements of the user observed by a tracking system provide valuable implicit and explicit input to the system. This input was used to derive a user model for performing tasks in the physical environment of a warehouse supported by pro-active assistance of the system. In particular navigation tracking and head-orientation sensing were used to derive cues of need for assistance. The aspects to be taken into account were reflected in the description of the system architecture for the MICA project, which has been implemented and validated for a warehouse application. The simple finite state machine identifying situations where pro-active assistance is needed was successful in improving effectiveness and efficiency in the picking process of a warehouse.

7. ACKNOWLEDGMENTS

Our thanks to the project team MICA and the support grant given by SAP research for the work performed in the project.

8. REFERENCES

1. Schilit, B.N., M.M. Theimer, and B.B. Welch. *Customizing mobile applications*. in *Proceedings of USENIX Symposium on Mobile and Location-independent Computing*. 1993. Cambridge, MA, US.
2. Dey, A.K. and G.D. Abowd, *Towards a better understanding of Context and Context-Awareness*. 1999, College of Computing, Georgia Institute of Technology. Technical Report.
3. Schilit, B.N., N.I. Adams, and R. Want. *Context-Aware Computing Applications*. in *Workshop on Mobile Computing Systems and Applications*. 1994. Santa Cruz, CA,.
4. Gross, T. and M. Specht. *Awareness in Context-Aware Information Systems*. in *Mensch & Computer 2001*. 2001. Bad Honnef (Germany): Teubner.
5. Chen, G. and D. Kotz, *A survey of context-aware mobile computing research*. 2000, Computer Science Department, Dartmouth College: Hanover, New Hampshire.
6. Oppermann, R., *From User-adaptive to Context-adaptive Information Systems*. *i-com Zeitschrift für interaktive und kooperative Medien*, 2005. **4**(3): p. 4-14.
7. Cohen, P., et al. *Quickste: Multimodal interaction for distributed applications*. in *Proceedings of the Fifth ACM International Multimedia Conference*. 1997. New York, NY: ACM Press.
8. Eccher, C., et al. *On the usage of automatic voice recognition in a web based medical application*. in *Proceedings of ICASSP*. 2003. Hong Kong, China.
9. Conçales, L.M.G., R.A. Grupen, and A.A.F. Oliveira. *A Control Architecture for Multimodal Sensory Integration*. in *Proc. of XI Int. Conference on Computer Graphics and Image Processing (SIBGRAPI'98)*. 1998. Rio de Janeiro, Brazil.
10. Oviatt, S. *Mutual Disambiguation of Recognition Errors in a Multimodal Architecture*. in *Proceedings of CHI 1999*. Pittsburgh, USA.
11. Fern, A., et al. *A Decision-Theoretic Model of Assistance*. in *International Joint Conference on Artificial Intelligence*. 2007.
12. Horvitz, E., et al. *The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users*. in *Fourteenth Conference on Uncertainty in Artificial Intelligence*. 1998. Madison WI: Morgan Kaufmann Publishers.
13. Pfaff, G., *User Interface Management Systems*. 1985, Berlin: Springer Verlag.
14. Wilson, M.D., et al. *An Architecture for Multimodal Dialogue*. in *Proceedings of the second Vencon Workshop on Multi-Modal Dialogue*. 1991.
15. Zimmermann, A., A. Lorenz, and M. Specht, *Personalization and context management*. *User Modeling and User Adaptive Interaction*, 2005. **15**: p. 275-302.
16. Barnett, J., et al. *W3C Multimodal Architecture and Interfaces*. 2006 [cited; Available from: <http://www.w3.org/TR/mmi-arch/> 1.3.2007.
17. Zimmermann, A. and A. Lorenz, *Creating Audio-Augmented Environments*. *Journal of Pervasive Computing and Communication*, 2005. **1**(1): p. 31-42.