Pers Ubiquit Comput (2007) 11: 59–68
DOI 10.1007/s00779-006-0066-7

ORIGINAL ARTICLE

Jasper Lindenberg · Wouter Pasman
Kim Kranenborg · Joris Stegeman · Mark A. Neerincx

Improving service matching and selection in ubiquitous computing environments: a user study

Received: 22 November 2004 / Accepted: 12 March 2005 / Published online: 30 May 2006
© Springer-Verlag London Limited 2006

Abstract In large ubiquitous computing environments it is hard for users to identify and activate the electronic services that match their needs. This user study compares the newly developed *service matcher* system with a conventional system for identifying and selecting appropriate services. The study addresses human factors issues such as usability, trust and service awareness. With the conventional system users have to browse a hierarchical list of currently available services and activate the service that they think satisfies their current needs. With the *service matcher* users just enter their current need using natural language, after which a wizard, emulating an existing service matcher algorithm, searches for and activates a matching service based on the given need and the users' location and gaze direction. This study shows that with the hierarchical list, only 66% of the tasks are solved correctly, and females score significantly worse than males. With the *service matcher*, the performance increases significantly to 84% correctly performed tasks and the gender difference disappears.

Keywords Service matching · Service awareness · User study · Ambient intelligence · Agents · Gender differences

1 Introduction

Living, travel and working environments contain a growing number of electronic services linked to networked devices and appliances (e.g., domestic services, travel-planning, entertainment and health-care services). An increasing number of those services will be non-stationary, such as services linked to cars, personal devices, mobile shops and movable furniture and appliances. Availability of services to a user may vary over time, depending on the user's position, network availability, access device, etc. [1–5]. Also the type of electronic services can vary enormously, from services for local physical control of lighting and heating conditions in a room to global non-physical non-localized services for electronic banking. For ubiquitous computing, one of the challenges is to adequately address the large heterogeneity and dynamic nature of users, services and environments.

This variety and dynamics leads, by its very nature, to the problem of how to identify and activate the appropriate service within a, continuously changing, multitude of services. How can intelligent techniques be employed to support users in finding the service matching their current needs?

Many solutions have been proposed on the technical part of the problem: how does a computer or electronic agent find a relevant service in an ambient intelligent environment? This is the area of service discovery. An overview and discussion of the state of the art of service discovery can be found in [6]. This paper takes a different approach, it focuses on an empirical human factors perspective on service matching and selection by means of an experiment with a newly developed *service matcher*. Technical research on the *service matcher* continued and ran simultaneously with the human factors research, providing a mutual advantage. The current paper presents the results of the human factors study, the technical results are also presented in this issue [7].

In general, very little attention has been paid to the non-technical part of the problem, human service discovery. Part of the researchers seems to view a human as just another agent, ignoring the enormous difference in information processing and interaction style. Also, agents are usually applied for a limited set of tasks in a

J. Lindenberg (✉) · K. Kranenborg · J. Stegeman
M. A. Neerincx
TNO Human Factors,
Kampweg 5, Soesterberg, The Netherlands
E-mail: jasper.lindenberg@tno.nl
Tel.: +31-346-356264
Fax: +31-346-353977
URL: http://www.tno.nl

W. Pasman · J. Lindenberg · M. A. Neerincx
Delft University of Technology,
Mekelweg 4, Delft, The Netherlands

narrow scope, while humans perform a wide range of tasks in diverse dynamic contexts.

However, some work has been done in the area of 'human service discovery'. Usually keywords can be used to find relevant services, for example using UDDI [8]. There are two problems with keyword searches. First, the user would have to know the exact keywords of the service to find it. It might be possible to extend the keyword list using thesauri as was proposed for web searching (e.g., [9]) but because words can have many meanings this might add wrong keywords to the list. Second, even more seriously, searching for a lamp would result in an overwhelming amount of hits.

Balke and Wagner [10] propose a method to refine the search based on user requirements and preferences. However, the keyword search is still the first step for accessing services, and—for instance in the case of searching for a lamp—the system might be flooded with requirement checks. Furthermore, it seems that the user has to build complex queries to work with their system.

Coen et al. [11] have a single room equipped with about hundred agents, each one controlling a device. Every agent may listen to the user after the keyword "computer" is spoken by the user. Each agent has its own grammar, optimized for its domain. It individually monitors that part of the user's context relevant to him in order to determine whether it actually will listen or not. Using the context and natural language to find the required service is similar to the *service matcher* approach. One problem with the approach of Coen et al. is that it is unclear how the room would react on multiple users having different tasks at the same time; it seems that they would heavily interfere with each other. When several similar devices are close to each other and all 'hearing' the user, the user will have difficulties targeting a single device. If a vague request is posed, the user might be overwhelmed with responses. Furthermore, as with the approach of Balke and Wagner, their approach seems not to scale to large areas. Finally, the user has to know where the device is and be close to the device before he can address it. This last issue is a fundamental problem: many services do not have a natural, human size, visible and/or unique physical embodiment.

The Phoenix parser [12] is a speech parser. It aims at a single speech-based application covering multiple services simultaneously, such as flight planning, hotel booking and car rental. Every service has a 'frame' containing the input fields for the request (e.g., departure time and location for a travel planner). The Phoenix parser can try to fit a user's utterance to multiple frames [13]. Thus, frame fitting works as a kind of service selection mechanism. Real life tests showed that the frame-based approach is very robust and effective. However, this parser has never been intended for fitting a large number of frames (services).

None of the available approaches really supports users in finding services appropriate for the task or problem at hand in a satisfactory way. In Ref. [6] the *service matcher* is proposed as a solution. With the *service matcher* system users formulate their wishes and needs in natural language and a system matches these needs to an appropriate agent (providing the service) in a context sensitive manner. Natural language enables the user to accurately describe this without the need of a priori knowledge. Automatic interpretation of natural language requests can be done robustly, if the context of the request (type of service targeted, location, etc.) is highly restricted. Therefore, in this system the parsing and understanding of the user's command is done locally by every service, instead of having a single parser/translator that would need to know every possible service in the system.

For accurate selection of the desired service, the service matcher is *context sensitive*. It takes into account information about the physical context of the user, the tasks the user is performing, the task history, the user's gaze direction, the user's location, etc. For instance, knowing the position of the user, the question "give me some light" would then only refer to the 'light' services bound to the particular room the user is in. Knowing the previous user requests could also provide a cue about the new question. Only if a question can not be addressed by a local or recently engaged service then the service matcher de-focuses and searches in a wider circle.

Once a service has been found that matches the user's request, the system automatically connects the user to the appropriate interface of the selected service. The agent that provides the selected service may handle the request itself, but it may also translate the user's request into several requests to other agents that all handle a part of the task. The *service matcher* system is assuming at least a *mobile* ad hoc *network* infrastructure for communication purposes. Agents running somewhere on that (possibly mobile and/or ad hoc) network infrastructure provide services. In such an infrastructure, agents (thus, the services) may enter and drop out of the system at any time [6, 7].

This paper describes an empirical study in which we compared the *service matcher* system to the traditional approach of manual browsing a hierarchical list of available services. For this comparison, the effects of two user characteristics will be explored because they might induce different individual preferences for a user interface. The first characteristic is gender as suggested by Marcus [14]. The second characteristic is locus of control, a characteristic that refers to the extent to which individuals believe that they can control events that affect them. Individuals with a high internal locus of control believe that events result primarily from their own behavior and actions. Those with high external locus of control believe that powerful others, fate or chance primarily determine events [15, 16].

To be able to assess the effectiveness, efficiency, trustworthiness, user satisfaction and service awareness of both approaches an intelligent environment was needed that contained a multitude of all types of services.

2 Intelligent environment

A (simulated) intelligent environment was created to enable this study. The environment contained agents representing and providing the services. For this study, we specified a 'day-in-the-life-of' scenario for the fictional main character (a role to be played by the participants). The spatial orientation of the agents was based on a part of Amsterdam were this fictional character was supposed to live and work. Three rooms in this environment were also physically modeled: the living room of his home (Fig. 1), the office room at his work (Fig. 2), and the gallery room of a museum.

The multitude of agents in the environment provided all types of services, such as lighting, AV-services, communication, heating, weather information, travel arrangements, alarms, security camera's, scheduling, etc. The agents were designed before the exact scenario for the test was made, to avoid bias in the agent structures. In total there were 552 agents in the environment, plus a few agents that could be created during the experiment, when needed. Roughly 10% of the nodes were related to the user (e.g., email, agenda), 20% to the home environment (e.g., lighting, AV-services), 20% to the office environment (e.g., security, coffee), 20% to the museum environment (e.g., information services, museum shop, lighting) and the remaining 30% to other services within Amsterdam (reservation, travel information). Figure 3 shows a small detail of the agents in the home environment.

The design of the agent space was based on the agent architecture described in [6]. However, as mentioned before, for our user study the agent system was a simulated intelligent environment so all 'agent intelligence' (the actual service matching) was done by a human wizard simulating the actual algorithm. Only extensive user interfacing support was implemented to help the wizard launch the appropriate interfaces quickly and to log the necessary experimental data (such as the user's location and gaze direction). The human wizard also provided the effects, such as actually switching on the light, the heater or coffee maker by means of an RF-remote, suggest and project a movie suiting a certain

Fig. 1 Living room of the home environment

Fig. 2 Office room

mood, provide a weather report or give information on a certain painter (in the museum). The wizard had access to the vocabulary of each agent but he did not use these during the experiments. The users could activate all 552 agents, but only 130 agents were implemented fully including complete user interface and actual effects. If one of the remaining agents was selected, the empty interface of that agent was shown, only containing the message 'the * agent has not been implemented' (* designating the name of the agent).

Two types of interfaces were available to activate a service in the intelligent environment (the two conditions for the actual study). The 'ask interface' is the natural language interface of the *service matcher* (Fig. 5a). The 'select interface' represents current common practice for selecting agents which enables the user to browse a location based hierarchy and to do a text-based search on (parts of) agent names (Fig. 4a). Both interfaces were used on a WLAN enabled mini-laptop (Sony VAIO

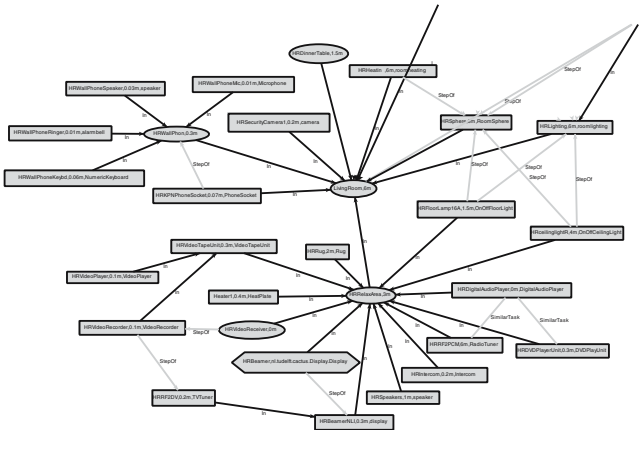

Fig. 3 Small part of the agents around the living room and their relations. 'In' relations indicate a spatial relation where agents fall within the service area of another agent. 'StepOf' relations indicate task relations, indicating that an agent may work as a subtask of another agent

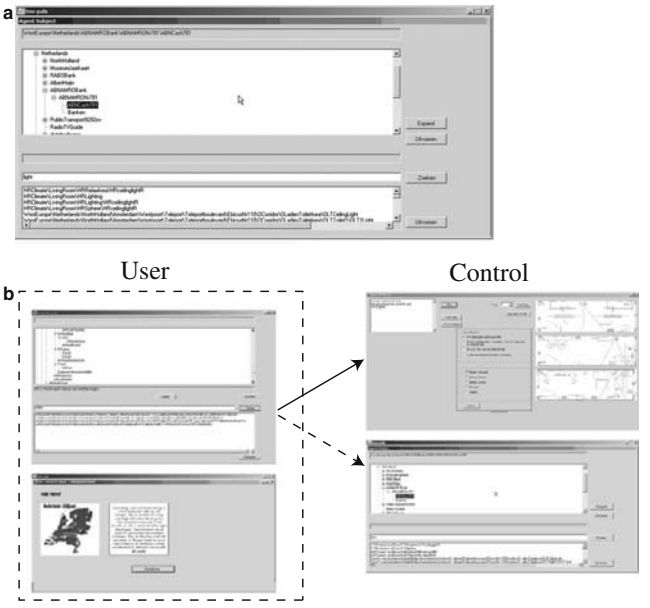

Fig. 4 **a** The select interface. Users select an agent using a hierarchy containing all agents. The hierarchy reflects the spatial location hierarchy of the objects related to the agents. **b** The interaction sequence for the 'select interface'. The user selects a service from the list and activates it. Information about the dialogue is sent to the control interfaces for logging and real-time monitoring purposes

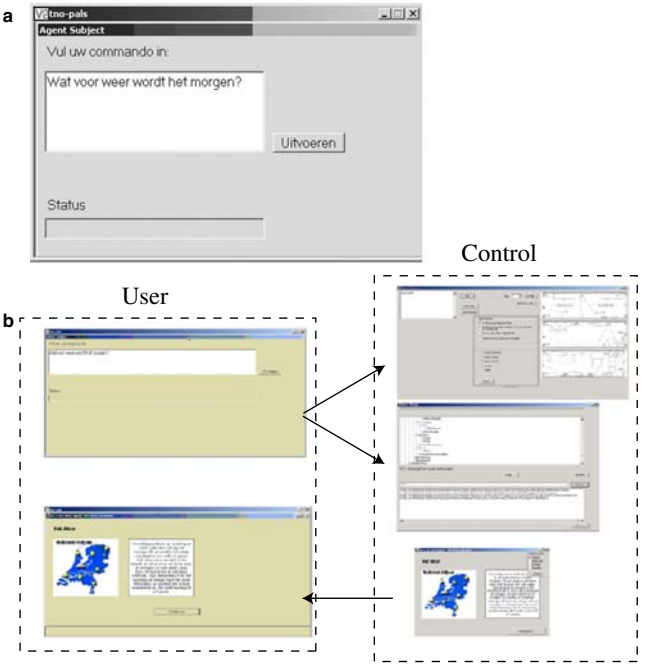

Fig. 5 **a** The ask interface. Users can type their needs and wishes in natural language. In this case 'What is the weather forecast for tomorrow?'. **b** The interaction sequence for the 'ask interface'. The user types a request in natural language. The request is sent to the control interface. The Wizard of Oz simulates the service matcher algorithm and activates the appropriate agent. Then, the Wizard of Oz prepares the agent and sends the interface to the user. The dialogue is logged for analyses

PCG-C1VE), which was a compromise between mobility and having a good keyboard.

With the 'select interface', users have to find an agent matching their needs by browsing through a hierarchical list of agents (Fig. 4a). The hierarchy reflects the spatial relationship between the agents in the system. As noted before, the names of the agents were chosen by a single person, and therefore were quite regular. In a larger multi-supplier ad hoc system without central management, searching in agent names probably would be even more difficult. An interaction sequence for the select interface is shown in Fig. 4b.

With the ask interface, the user types his request in natural language using the keyboard on the mini-laptop (Fig. 5a). The wizard receives the request, and tries to find a matching service. The wizard uses his knowledge about the user's context, agents in the environment, recently used services, the user's location and gaze direction (using a closed circuit television), etc., to make a match. The wizard tries to stick to the match finding algorithms developed for the service matcher, but obviously this is only an approximation of what a computer would do. The wizard can activate a service for the user, present a list of services to choose from to the user (in the case of multiple matches), or give a failure message if a suitable match was not found.

With the ask interface, if an agent is selected, a template of the user interface for that agent is opened on the wizard's control screen. The wizard can adjust the template according to specifics that the user asks (for instance, if he asked to start the fine wash at 12 o'clock, the wizard could already pre-select this option in the interface, see Fig. 6. The (usually partially) instantiated service interface is then sent to the user for further specification and acceptance (Fig. 5b). In the 'select interface' condition the service interface is directly opened on the user's device (Fig. 4b).

3 Experimental setup

The goal of the study was to make an assessment of the differences in performance, usability, trust and service awareness between the two interface types (the 'ask' and the 'select' interface) representing the *service matcher* and the 'classic' approach to service selection.

The experiment consisted of a between subjects design with interface type as the independent variable. Dependent variables were the number of correctly performed tasks (effectiveness), total session time (efficiency), changes in emotional state, trust, service awareness and subjective usability (satisfaction). Specific attention was given to gender and locus of control as predictors.

A task was judged to be *performed correctly* if the participant had reached an agent that was appropriate for the task at hand. How the agent was exactly instructed or manipulated by the participant after that point did not matter for the judgment.

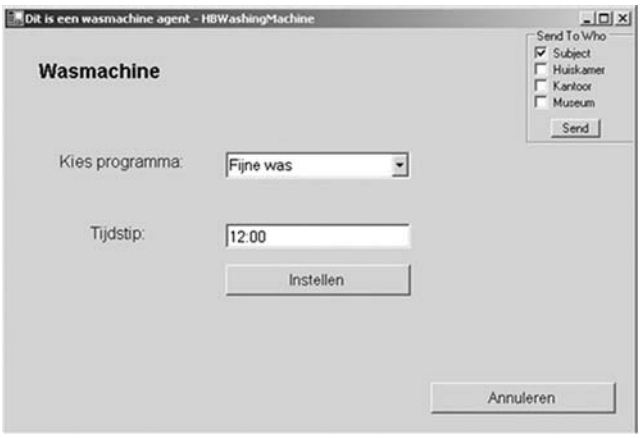

Fig. 6 Template for the interface of the washing machine agent, instantiated by the wizard for fine wash at 12 o'clock. When the wizard clicks the 'Send' button the interface appears to the user

Trust was measured using a questionnaire with three propositions based on a questionnaire described in [17]. The propositions are (1) This system helps you to exactly find the information and perform the actions that you wish (2) This system offers you the possibility to find information and/or execute actions, every time and in an adequate and consistent way (3) I can trust this system. The participant could rate each proposition with a number between 1 (totally inapplicable) and 7 (very applicable). More details can be found in [17].

The *locus of control* was measured using a questionnaire, and is supposed to be constant for a person. It contains 20 questions like "Usually I get what I want in life", each to be answered with "agree" or "disagree". Each question adds 0 or 5 points to the score, for a final score between 0 (a person feels to have no control at all) and 100 (the person feels everything is under control) [15, 16].

The *subjective usability* was judged with another questionnaire. It contained eight propositions such as "I understand the behavior of this system", which can be rated with "I agree" or "I disagree". Also the user was asked to give three strong and three weak points of the system.

Measurement of *emotional state* was done using the Self-Assessment Manikin (SAM). This subjective scale is based on the pleasure arousal dominance (PAD) model of emotion and describes emotion on the three dimensions of valence, arousal and dominance [18]. The scale presents three rows of cartoons, on which the subject has to characterize the experienced emotion. In earlier studies the dominance scale proved to explain the least variance, and had the highest variability in terms of its inferred meaning. Therefore the dominance scale is omitted for the present experiment.

Service awareness was measured after the experiment. Service awareness is a measurement, to assess the quality of the expectations users have of the available services in a certain environment (do they know which services are available in a certain environment). In this study it was measured by asking the participants to list all the services available in a certain environment (living room, office and gallery). It was made clear to them that it was not necessary to have used the service in the scenario. The participants were asked to write down as many as possible services in the living room, office and gallery. The score is the number of recalled services.

3.1 Participants

The participants were 8 female and 11 male students from Utrecht University. They were 18–28 years old, 23 years on average. From the personal background questionnaire we found that they mostly followed a non-technical study (history, law, psychology, etc.). They had good experience with working with the Internet, their experience with computers and mobile devices was about average. The participants were paid for their participation.

3.2 Procedure

The sessions were organized as follows. On arrival, the participant started with an introduction provided by the test leader and was asked to sign a consent form. Then the participant received instructions on how to operate the assigned interface, some pitfalls, and a short explanation on context sensitivity. Next, they had to fill in the questionnaires on locus of control and a questionnaire on their personal background and technology experience. Then, the experiment started, where the participant followed a scenario given to him on paper containing 32 'tasks'. At this point the test leader went to the control room. Audio and video of the test rooms was available in the control room for purposes of monitoring and performing the wizard role. Six times during the experiment, they had to fill in the SAM questionnaire. After completion of the scenario the trust, usability and service awareness questionnaires were filled in. During the service awareness test, participants were allowed to go back to the rooms to enhance their recall. In total, a session including instructions, questionnaires and payment took about 2.5 h.

3.3 Scenario

The participants had to follow a scenario, leading to tasks that had to be fulfilled. The scenario described a day in the life of Jaap Kal, a married small business owner working and living in Amsterdam. Direct task instructions were avoided because that would prime the participants with words needed to find relevant agents. Each scenario session contained 32 *tasks* to be performed by the participant (e.g., a task could be "set a comfortable atmosphere"). To complete a task, the participant had to interact with one or more agents.

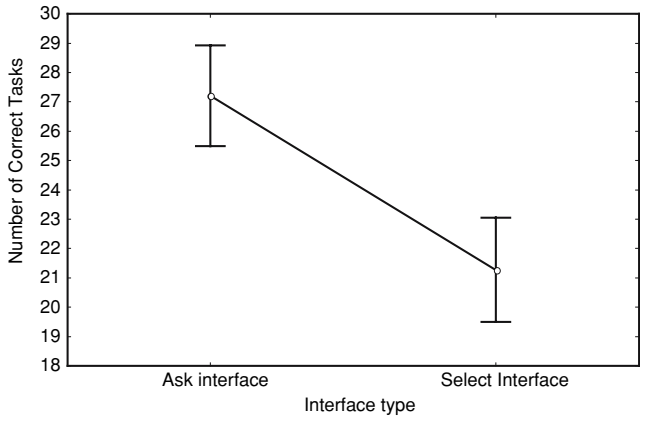

Fig. 7 Effect of interface type on the number of correctly performed tasks (*bars* denote a 95% confidence interval)

Often, there were multiple correct solutions to perform a task. The first 16 tasks were done in the living room, the subsequent 11 tasks were in gallery, and the final 5 tasks in the office. Every attempt of the participant to get contact with an agent—either using a natural language request or a selection of an agent—is called a *trial*. Multiple trials could be used for a single task.

4 Results

All data was analyzed using StatSoft Statistica and was checked for normality using Kolmogorov–Smirnoff test for normality and Shapiro–Wilks W test.

The *main result* of the experiment is the effect of the interface type on the number of correctly performed tasks. With the ask interface participants successfully completed on average 27 tasks (84%), while they completed only 21 tasks (66%) with the select interface (Fig. 7). An ANOVA showed this to be a highly significant effect [$F(1,15) = 26.169$, $p < 0.001$].

There was an effect of gender on the number of correctly performed tasks. Males completed more than 25 tasks correctly on average, while females complete only 23 tasks [$F(1,15) = 4.584$, $p = 0.049$], see Fig. 8.

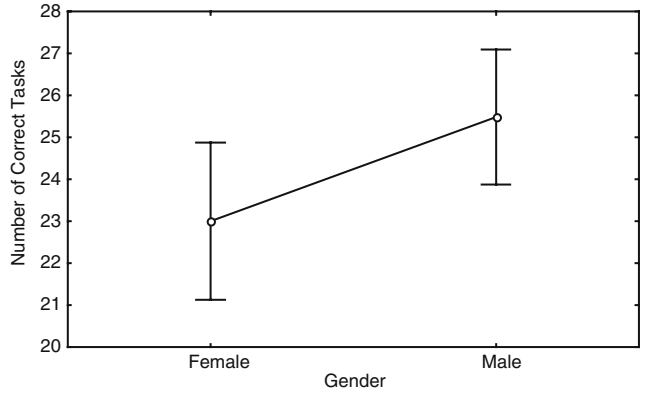

Fig. 8 Effect of gender on the number of correctly performed tasks (*bars* denote a 95% confidence interval)

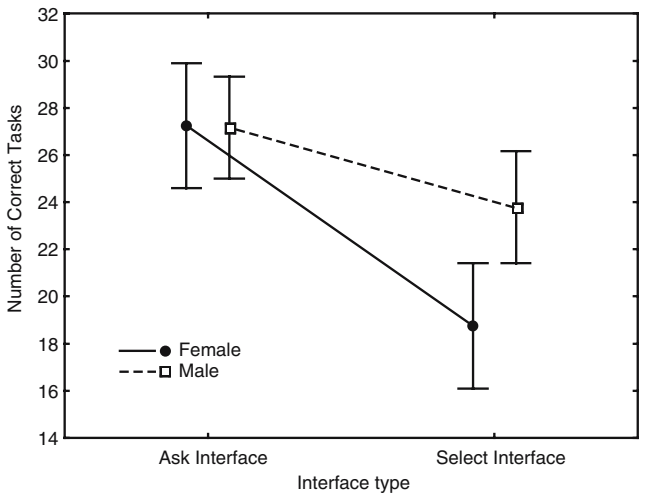

Fig. 9 Effect of interface type and gender on number of correctly performed tasks (*bars* denote a 95% confidence interval)

Further analysis (Fig. 9) shows a significant interaction effect between gender and interface type [$F(1,15) = 4.897$, $p = 0.048$]. Tukey HSD post hoc analysis shows that males perform better with the select interface than the females [$F(1,15) = 6.209$, $p = 0.039$]. Gender does not have an effect on performance with the ask interface.

Locus of control did not have a significant effect on the number of correctly performed tasks. No significant effects of interface type on total session time was found (this is probably caused by the time-limit that was set for the different tasks). However, the interaction between interface type, gender and total session time shows the same trend as in the interaction between interface type, gender and number of correctly completed tasks, the performance of the females drops much more with the select interface than the males.

There was no difference in trust scores between the two interface types. The trust scores for the ask interface condition were 4.73 on average, with a standard deviation of 0.73, the scores for the select interface were 4.70 with a standard deviation of 0.92. These scores are relatively high concerning the fact that in earlier experiments comparable values on trust were found for working with a standard website via a laptop [19]. It was expected that trust would increase when the participant performed well. However, no direct effects of the number of correctly performed tasks on the trust scores were found. No effect was found on the usability scores. However, a goodness of fit ANOVA showed an effect of usability scores on trust scores ($F(1,17) = 8.353$, $p < 0.010$ (Fig. 10), resulting in the following regression equation ($R^2 = 0.329$):

$$\text{TRUSTSCORE} = 11.548 + 0.574 \times \text{USABILITYSCORE}.$$

The select interface seems to encourage people slightly better in recalling available services (Fig. 11), but this effect of interface type on service awareness is

not significant [$F(1,15) = 2.347$, $p = 0.146$]. Gender has a significant effect [$F(1,15) = 4.697$, $p = 0.047$] males seem to have a better service awareness (they are better at recalling services than females, Fig. 12). The interaction effect (Fig. 13) is not significant [$F(1, 15) = 2.3471$, $p = 0.14634$] and a Tukey post hoc analysis does not show any significant results. However, a Fisher LSD post hoc analysis shows that males with the select interface score significantly higher on service awareness than female-ask ($p = 0.021$), female-select ($p = 0.033$) and male-ask ($p = 0.021$); as shown in Fig. 13.

The effect of the locus of control score on service awareness was also analyzed. The locus of control scores were partitioned around the median, in a 'low' and a 'high' group. Locus of control scores do not seem to have an effect on service awareness at all [$F(1, 15) = 0.40785$, $p = 0.533$] (Fig. 14). However, a more detailed analysis (Fig. 15) shows the surprising role gender plays, the combined interaction is significant [$F(1,15) = 7.399$, $p = 0.016$]. A Tukey HSD post hoc analysis shows that for participants scoring high on the locus of control, males have significantly higher service awareness than females [$F(1,15) = 19.860$, $p = 0.021$]. For participants with low locus of control score, the gender makes no difference for the service awareness.

The services awareness of the participants remained limited to a recall of the services they had actually used during the experiment. They apparently do not remember agents they encountered but did not use and they also did not extrapolate new services from the ones they used.

We checked for effects on the number of words used by the participants in the ask-interface. Neither gender nor locus of control seems to have an effect on the number of words used. Also there is no effect of time (trial number) on the number of words used. There were a few peaks in the number of words used for some trials, but those seem more related to a few hard tasks. The changes in valence and arousal scores (emotion) did not show any significant effects.

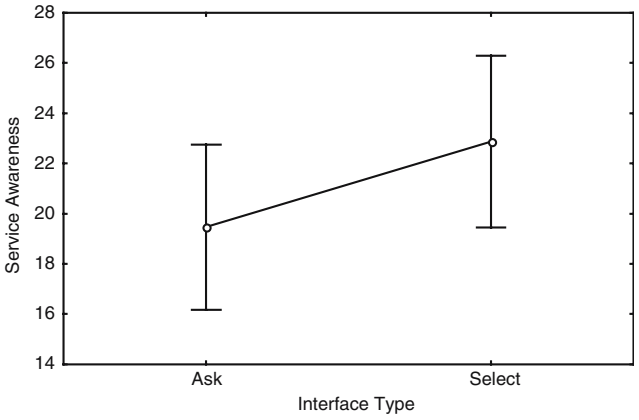

Fig. 11 Effect of interface type on service awareness (*bars* denote a 95% confidence interval)

5 Discussion

The results of the experiment were quite surprising. In the following sections we will comment on and take a closer look at the results.

5.1 Wizard of Oz

Based on the verbal comments made by the participants ('the system is too slow') and their behavior (they started working on the next task with the wizard/test leader still in the room with them) we can safely conclude that the Wizard of Oz manipulation worked. All participants believed that there was an actual system performing the service matching. Ironically, non-fatal system failure and inexplicable error messages seemed to increase the participants' belief in an actual system.

Using a Wizard of Oz method for this study certainly proved beneficial. We were able to perform an ecologically valid experiment with severely reduced programming costs and duration. Because of meticulous planning and execution of the Wizard of Oz illusion no negative side effects occurred. A detailed discussion on

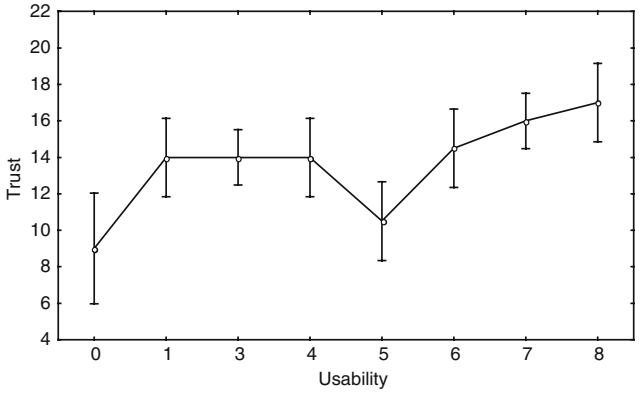

Fig. 10 Effect of usability on trust (*bars* denote a 95% confidence interval)

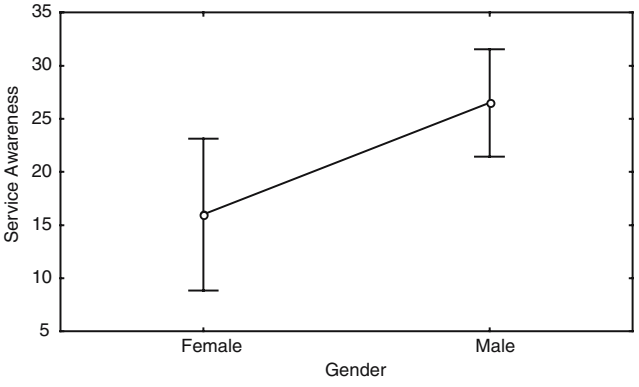

Fig. 12 Effect of gender on service awareness (*bars* denote a 95% confidence interval)

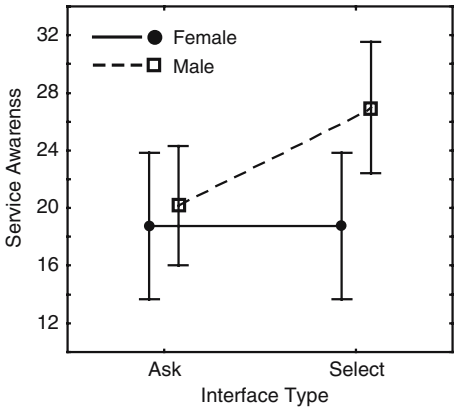

Fig. 13 Effect of interface type and gender on service awareness (*bars* denote a 95% confidence interval)

the advantages and disadvantages of the Wizard of Oz method including the ethical aspects can be found in [20, 21].

5.2 Effectiveness and efficiency

As shown in the results, the *ask* interface proved to be more effective than the *select* interface for identifying and selecting an appropriate service. Effects on efficiency (time) were not found. This is probably caused by the fact that a maximum task time was implemented (to limit the total session time). With a more lenient time limit there might not have been an effect on effectiveness but then there would probably have been an effect on efficiency instead.

The intelligent environment developed for this study was quite extensive but a real system would probably contain much more agents having a much wider variety of names. It is therefore to be expected that the performance difference between the *ask* and *select* interface would only increase further.

5.3 Validity

One might argue that the performance difference occurs because the select interface and its hierarchy are not optimally designed. This is probably true, but for our case, ad hoc networks without central management the hierarchy would normally be much worse. A well-designed hierarchy combined with a select interface might be more effective for small networks with centralized control and management. However, in our opinion, this does not seem to be the direction in which ubiquitous networks are moving [22, 23].

5.4 Lostness and service awareness

As expected, the select interface caused orientation problems for the participants. They got lost in the large agent hierarchy, and because of this they frequently failed to complete tasks. This disorientation is probably the main cause for the reduced performance of the select interface. However, when given unlimited time the large agent hierarchy of the select interface also provides an advantage. With the select interface, if all other strategies fail, every interface could be activated in a sequential manner until an appropriate one is found. As a number of participants mentioned, this cannot be done with the ask-interface. If you are unable to come up with the 'right' words to describe your needs, a dead-lock situation can occur. Also the results seem to point at improved service awareness with the select-interface (Fig. 13). This seems likely because the select interface can show the availability of services that the user would not think to ask for because their existence is not expected. Nevertheless only males seem to benefit, probably caused by an effect treated in the section on gender.

Some problems related to service awareness occurred because of varying availability of higher-level agents. For instance, in the office there was no agent managing overall room lighting. Therefore requests like "turn off all lights" would not be understood at that location, while it was understood in the living room. The fundamental problem seems to be how the user can know the 'intelligence level' of his current environment. We think this problem is critical for smart environments and needs more research. A solution might be the development of explanatory interface components [24] for intelligent environments.

In retrospect the definition and measurement of service awareness was not ideal. Both definition and measurement focused too heavily on memory. To refine the definition and improve the measurement techniques we suggest using the concept and measurement techniques of situation awareness [25] and its three levels (perception, comprehension and projection) as a solid basis.

5.5 Gender

Many of the effects in this study were mediated by gender, other recent experiments have found similar effects. Especially when tasks and interfaces demand navigating hierarchical structures, males seem to perform better. Studies suggest that this effect is related to the cognitive factor spatial ability [26]. Based on our results gender based interface adaptations [14] seem a good idea especially since current personalization and automatic interface generation techniques provide the means for realizing these interfaces [27].

5.6 Use of user location and gaze direction data

All sessions were recorded on video for analysis of movement patterns of the participants. However, the participants hardly moved at all, usually they just sat down somewhere with the mini-laptop. We think that the device used in this study is still too large for input

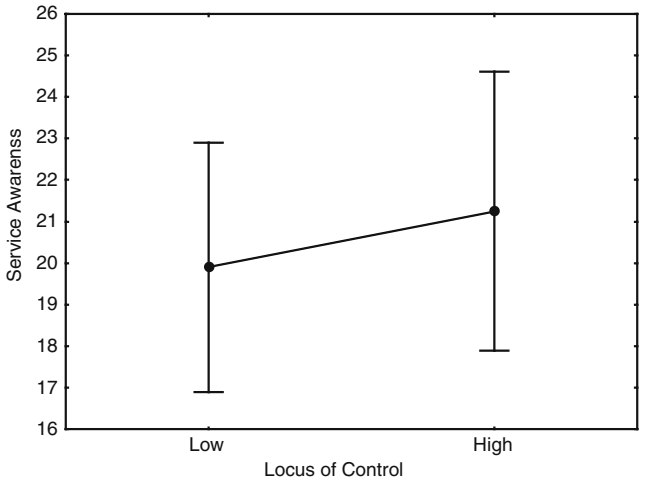

Fig. 14 Effect of locus of control on service awareness (*bars* denote a 95% confidence interval)

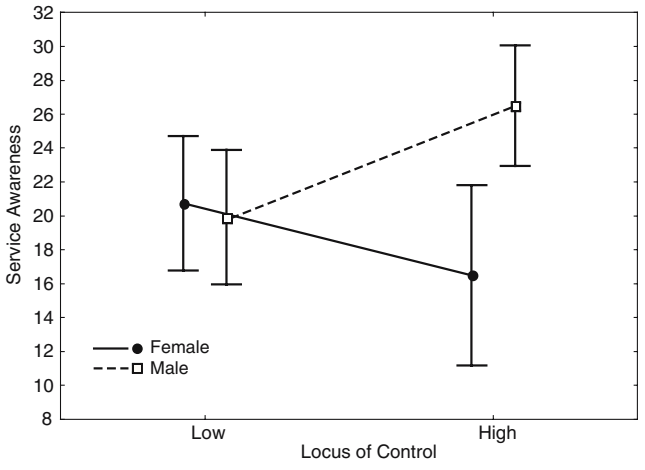

Fig. 15 Effect of locus of control and gender on service awareness (*bars* denote a 95% confidence interval)

while walking, and also the keyboard input device promotes sitting down to correctly place your hands for typing. This relative immobility can probably also be improved by using speech as the primary input. Also the rooms in which the experiments took place were relatively small (about 15 m^2 per room). Larger areas will probably promote mobility.

The gaze direction that is used by the service matcher algorithm was less useful than anticipated. The wizard logged the gaze direction and integrated this in his simulation of the *service matcher* algorithm. We hoped that the subjects would, for example, look at a painting before typing "who is the artist?". However, in the experiment people mainly looked at the mini-laptop or the instructions. This can be partly blamed on the experimental setup but that does not seem to be all. The user interfacing would need serious rethinking to make gaze direction an effective input parameter, even when using speech as primary input (in that case users will probably look at the suspected location of the microphone instead).

Because of the limited mobility and the aforementioned problems with gaze direction, it is difficult to reflect on the users' perception of the context awareness of the *ask* interface. However, observations by the test leader indicated that context awareness proved to be a difficult concept for users. The participants seemed to understand the principle of context awareness when they walk to or look at a light before asking to turn on the light. But, a minute later they have difficulties targeting a specific painting for information. It is not yet clear what causes this effect but it seems to be related to the level of abstractness of the service (a physical light compared to information about a painter linked to a painting).

5.7 Emotion, trust and usability

As indicated in the section on the experimental setup we also looked at changes in emotional state (valence and arousal) during the experiment. We did not find any effects of the interface type, gender or locus of control on an emotional state. Our accumulated impression of several experiments measuring emotion [19] is that it is difficult to influence the emotional state unless the experiment is explicitly designed to do so.

As indicated in the results section the trust scores were relatively high for such a novel and dynamic application. An interesting trust issue observed afterwards is that (with the *ask* interface) people sometimes copy earlier requests to the letter, in order to reach the same agent they had reached before. Clearly, and maybe rightfully, they are afraid that even the slightest change could change the behavior of the system. This seems to suggest that the users trust that the system will behave in a consistent manner but they do not trust it at a higher level (that the system will enable them to find the right service even without remembering the exact request). It is an interesting topic for further research.

The usability scores did not provide any interesting results except the effect of usability scores on trust scores. However, this effect might be an artifact caused by a certain semantic overlap in both questionnaires. But besides the usability questionnaire user comments on both interfaces were also collected during the experiment. Almost all participants mentioned that the system seemed to be overkill for easy tasks such as switching on a light. Indeed, in many cases switches are more appropriate than typing natural language, but this was outside the scope of the current research. Almost all participants indicated that the system was an ideal tool for lazy persons, ironically about half the participants listed this comment under 'negative aspects' and the other half under 'positive aspects'.

A minority mentioned that they did not like to be so dependent on a single device to control their life and environment so back-up functionality should always be available. Also the participants in the ask-interface condition mentioned that the level of natural language recognition was too limited and could be improved. This

can actually be improved by training the system with actual requests as was done with the data gathered in this experiment [7].

6 Conclusions and future work

The *ask* interface in combination with the *service matcher* algorithm seems to offer a promising alternative solution for user based service selection. Users were better able to find the services they needed than with the classic hierarchical list approach represented by the *select* interface. A positive aspect of the *select* interface is that it is suited for preventing the dead-lock situation that might occur in the *ask* interface (if the user is unable to express his needs or runs out of words or phrases formulating the issue). Also the select interface can show the availability of services that the user would not think to ask for because their existence is not expected. A solution seems to be to use the ask interface for general use and integrate a select interface component which can be activated in dead-lock situations or to discover unsuspected services. In this study gender seemed to play a significant role suggesting that personalization of interface features based on the user's gender might be a good idea. Both the integration of the *ask* and *select* interface, the refinement of the *service matcher* algorithm and the gender based interface adaptations are considered future work. Another issue for future research is the development of interfaces that support the service awareness, context awareness and assessment of 'intelligence levels' of environments by users.

Acknowledgments This work is part of the CACTUS project which was co-funded by the Dutch Ministry of Economic Affairs as part of the policy plan "Concurreren met ICT Competenties".

References

1. Aarts E, Harwig R., Schuurman M (2001) Ambient intelligence. In: Denning P (ed) The invisible future. McGraw Hill, New York, pp 235–250
2. Abowd GD, Mynatt ED, Rodden T (2002) The human experience. IEEE Pervasive Comput 1(1):48–57
3. Fogg BJ (2003) Persuasive technology: using computers to change what we think and do. Morgan Kaufmann, Amsterdam
4. Gajos K, Fox H, Shrobe H (2002) End user empowerment in human centered pervasive computing. In: Proceedings Pervasive 2002, Zurich, Switzerland, pp 134–140
5. Satyanarayanan M (2001) Pervasive computing: vision and challenges. IEEE Pers Commun 8(4):10–17
6. Pasman W (2004) Organizing ad hoc agents for human-agent service matching. In: Proceedings Mobiquitous 2004, Boston, MA, pp 278–287
7. Pasman W, Lindenberg J (2006) Human-agent service matching using natural language queries: system test and training. Pers Ubiquitous Comput (this issue)
8. Universal description, discovery and integration of web services. http://www.uddi.org
9. Wang Y, Stroulia E (2003) Semantic structure matching for assessing web service similarity. In: Proceeding international conference on service oriented computing, Trento, Italy
10. Balke W, Wagner M (2003) Towards personalized selection of web services. In: Proceedings of the 12th international world wide web conference, Budapest, Hungary
11. Coen M, Weisman L, Thomas K, Groh M (1999) A context sensitive natural language modality for an intelligent room. In: Proceedings of 1st international workshop on managing interactions in smart environment, Dublin, Ireland, pp 68–79
12. CSLR (1991). The Phoenix parser user manual. http://www.communicator.colorado.edu/phoenix/Phoenix_Manual.pdf
13. Constantinides PC (1999) Scoring techniques for Phoenix Parses. Available as part of the Phoenix documentation. http://www.fife.speech.cs.cmu.edu/Phoenix
14. Marcus A (1993) Human communication issues in advanced user interfaces. Commun ACM 4(4):101–109
15. Rotter J (1966) Generalized expectancies for internal versus external control of reinforcements. Phychol Monogr 80(609)
16. Marsh HW, Richards GE (1986) The rotter locus of control scale: the comparison of alternative response formats and implications for reliability, validity and dimensionality. J Res Pers 20:509–558
17. Jian J, Bisantz A, Drury C (2000) Foundations for an empirically determined scale of trust in automated systems. Int J Cogn Ergon 4:53–71
18. Bradley M, Lang P (1994) Measuring emotion: The Self-Assessment Manikin and the Semantic Differential. J Behav Ther Exp Psychiatry 25:49–59
19. Neerincx MA, Streefkerk JW (2003) Interacting in desktop and mobile context: emotion, trust and task performance. In: The EUSAI conference proceedings, Eindhoven, Netherlands
20. Dahlback N, Jonsson A, Ahrenberg L (1993) Wizard of Oz studies—Why and How. In: The proceedings of the 1st international conference on intelligent user interfaces, ACM, New York
21. Fraser N, Gilbert NS (1991) Simulating speech systems. Comput Speech Lang 5:81–99
22. Foster I, Kesselman C, Nick J, Tuecke S (2002) The physiology of the grid: an open grid services architecture for distributed systems integration. open grid service infrastructure WG, global grid forum. http://www.globus.org/research/papers.html
23. Beute B (2002) Navigating distributed services. Doctoral Thesis, Center for Tele-Information, Technical University of Denmark. http://www.cti.dtu.dk/publications/phdthesis.view.php?id = 17336
24. Paymans TF, Lindenberg J, Neerincx MA (2004) Usability trade-offs for adaptive user interfaces: ease of use and learnability. In: The proceedings of intelligent user interfaces, Funchal, Portugal, pp 301–303
25. Endsley MR (1995) Toward a theory of situation awareness in dynamic systems. Hum Factors 37:32–64
26. Neerincx MA, Lindenberg J, Pemberton S (2001) Support concepts for web navigation: a cognitive approach. In: The proceedings of the 10th international WWW conference, Hong Kong, pp 119–128
27. Gajos K, Weld DS (2004) SUPPLE: automatically generating user interfaces. In: The proceedings of the 8th international conference on intelligent user interfaces, ACM, New York