Voice-based Error Recovery Strategies for Pervasive Environments *

[Extended Abstract] †

Dirk Schnelle-Walka
TU Darmstadt - Telecooperation Group
Hochschulstr. 10
64289 Darmstadt, Germany
dirk@tk.informatik.tu-darmstadt.de

Stefan Radomski
TU Darmstadt - Telecooperation Group
Hochschulstr. 10
64289 Darmstadt, Germany
radomski@tk.informatik.tu-darmstadt.de

Arvid Lange
TU Darmstadt - Telecooperation Group
Hochschulstr. 10
64289 Darmstadt, Germany
lange@stud.tu-darmstadt.de

ABSTRACT

Errors in speech recognition systems severly hinder users from controlling their environment by voice since the interaction with the system usually relies on the user to repeat the command until it is successful. Although error recovery strategies are well known and understood in telephony environments they were not adopted in command & control scenarios. In this paper we introduce a system that utilizes these techniques for a better user experience in which the interaction is perceived as a dialog.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces And Presentation**]: User Interfaces—*Voice I/O, Input devices and strategies, Prototyping*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

General Terms

Design, Human Factors

Keywords

Error Recovery, Voice User Interfaces, Smart Spaces

1. INTRODUCTION

Controlling devices in smart home environments, like light, shutters or television is already possible with off-the-shelf solutions. Usually, this is achieved with the help of wall-mounted touch screens at a central location in the home. Mobile solutions utilizing smart-phones or tablets are also available. However, these devices require users to pick them up and carry them along. Especially in home environments users will usually place them either at a certain location or will even misplace them, encountering difficulties in finding them when needed. As a matter of fact, it will be unlikely that users will have the device at hand when it is needed. Speech on the other hand is always available and a lot of research focused on enabling the user to speak freely without explicit devices and hence use a more natural form of interacting with pervasive environments. However, the fact that speech cannot be recognized with a recognition rate of 100%, usually remains a problem in such command & control scenarios. In most cases the user is forced to repeat the command until it is either correctly recognized or she gave up. This severely hinders the successful application of voice based interfaces in these environments.

Error correction or recovery strategies that are known from telephony based systems are not regarded for command & control settings. In this paper we describe our experiences while applying error correction strategies to a voice based system to control smart homes, already described in [11].

2. RELATED WORK

One of the available off-the-shelf solutions is Gira Speech Control[1,2]. The system can record up to 64 different commands to select a menu item or to execute an action. The selected action is executed at the HomeServer which acts as a gateway to the KNX/EIB installation. It is also possible to utter multiple commands in a row, like *"Dim dinner table to 80 percent, dim couch to 80 percent"* to execute more than a single command at once. However, this does not appear very natural. A more natural interaction would also allow for inputs like *"Dim dinner table and couch to 80 percent"*. Moreover, in case of an error the system simply rejects the input and the user will have to input everything anew.

Other systems try to improve the situation by giving hints to users as to what they might say in the current context. Sagawa et al. [9] describe an agent based system that aims

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

†A full version of this paper is available as *Author's Guide to Preparing ACM SIG Proceedings Using LaTeX2ε and BibTeX* at `www.acm.org/eaddress.htm`

MobileHCI Simpe '13 Munich, Germany

[1] `http://www.gira.de/produkte/homeserver.html`
[2] `http://www.youtube.com/watch?v=4TP1dPvUzWc`

at minimizing the number of dialog turns with the help of correction grammars. Here, a user initiated error correction results in a confirmation of the correction in order to return to the normal dialog flow as soon as possible. E.g., in case the system falsely understood *Kyoto* instead of *Tokyo*, the user may interrupt the current prompt by stating e.g. *"I said Tokyo"*. Technically this is achieved by adding a generated correction grammar for possible user corrections to the next slot. This way, they were able to reduce the number of dialog turns.

Newer systems like INSPIRE [8] introduce an error taxonomy for a better error handling. They distinguish between errors at the (i) goal-level in case the capabilities of the system are misunderstood, (ii) task-level in case the user does not understand how to achieve his goal for interacting with the system, (iii) command-level for vocabulary and grammar errors and (iv) conceptual errors if the user refers to the world in a way that is not understood by the system. The authors introduce several solutions for the different error types like ESCALATING DETAIL [10] which is also known as tapering [1, 3] for conceptual level errors. However, they provide no general concept how to cope with errors but state that *there is no "silver bullet" that can lead to great progress...but when we focus on a particular type of problem in a particular type of system, specifically applicable solutions can often be identified* [8].

Novel concepts like OwlSpeak [5] use ontologies to generate VoiceXML documents dynamically. However, they merely try to resolve out-of-vocabulary words to *avoid the necessity of commands to be repeated by the user"* [4]. In fact, they rely more on error prevention rather than error correction.

3. APPROACH

It has been shown that error recovery and error correction must be a fundamental part in the design of voice based applications [12]. This is also true for controlling the environment by voice. Therefore we hypothesize that the integration of error correction and error prevention will improve the user experience by enabling mixed initiative concepts in command & control settings. In these cases the system may ask for missing data as it would be necessary in the following dialog taken from [11].

User: Please close the shutter
System: Which shutter shall be closed?
User does not know how to continue and says nothing
System: Do you want to close the shutter to the garden or to the terrace?
User: The terrace

Therefore, we extended our application of a conversational approach to command & control settings that we described in [11]. For the error correction we settled upon known error recovery and prevention strategies as described in [10]. The strategies make use of a categorization of errors that was created by Duff [2]. For convenience a description of the categories is copied from [10]:

Level 0 Missing input.

Level 1 Recognition rejection.

Level 2 Recognizer returns something that cannot be interpreted (makes no sense at all).

Level 3 Recognizer returns something that is not semantically consistent.

Level 4 Recognizer returns semantically well formed, but impossible to fulfill sentences.

Level 5 Same as 4 with the exception that the impossibility is due to the dialog context.

Level 6 The back-end system fails to fulfill the command

Level 7 User initiated error correction.

Depending on the category there are several error management design patterns that can be used to handle the errors. Again, for convenience we provide thumbnails [6] of the patterns that are described in [10] and are mentioned in this paper:

Escalating Detail When a user's speech is not recognized correctly or the user did not speak at all, provide responses that give increasing amounts of detail with each subsequent error with respect to the source of the error. These responses should be designed to help the user in making less ambiguous verbal responses.

Rapid Reprompt: Same as ESCALATING DETAIL with the exception that the first response is very short assuming that the user knows what to say and simply needs another try.

Global Error Correction: Count the global amount of errors per type. If one of these counters is greater than a predefined threshold, transfer the user directly to an agent or name other means of achieving her goal for calling.

Selection From A List: In case there are multiple recognition results with a similar confidence score, use the hypothesis list to ask the user for confirmation of each of them using simple yes-no questions.

Three Tiered Confidence: Imitate the behavior of humans in a dialog who accept the data if the other party said something that was understood without any doubt but ask for confirmation if there was some uncertainty about words recognized. If the other party was not able to understand what was said at all, simply ask again. Therefore, establish three tiers for each of these cases and assign an error recovery strategy for each tier.

Implicit Confirmation: Mimics the behavior of the human to ensure the correctness of the entered data. Therefore, integrate the entered data into the subsequent output that asks for the next piece of information.

We suggest to use patterns for the different levels and error management design patterns as follows.

For Level 0, Level 1 and Level 2 errors we suggest to use ESCALATING DETAIL or RAPID REPROMPT. Also THREE TIERED CONFIDENCE could be applied to save as much as possible from the user's input.

An example dialog using the Rapid Reprompt pattern is the following:

User: Please close the *unrecognized input*
System: I did not understand you. Please repeat.
User: *unrecognized input*
System: I did not understand you. You can simply name the device to control and what to do with it.
User: *unrecognized input*
System: Sorry, I still did not understand you. Which device do you want to control?
User: The shutter.
System: Do you want to close the shutter?
User: Yes

In case of a Level 3 error and if no matching keyword could be detected Escalating Detail or Rapid Reprompt could be used. If a keyword like *shutter* is detected, the dialog could reuse that, e.g., by Selection From A List to continue in a meaningful sense to return to the usual dialog path. In this case the dialog could be as follows:

User: Please close the shutter
System: Which shutter shall be closed?
User does not know how to continue and says nothing
User: *unrecognized input*
System: I did not understand you. You can simply name the device to control and what to do with it.
System: Do you want to close the shutter to the garden or to the terrace?
User: The terrace.

For the error levels 4 to 7, the only viable solution could be to give a short explanation why the command could not be executed. E.g., the system could state that it tried to close the shutter but sending the command to the corresponding actor failed. In order to be helpful it should also suggest other means for reaching the user's goal.

A possible dialog in these cases could be the following.

User: Please close the shutter to the terrace
System: OK. Closing the shutter.
It is not possible to close the shutters.
System: I am sorry, but there seems to be an error with one of the shutter's motors. Please call a craftsman for repair.

For Level 7 errors there are no dedicated patterns but the system should maintain a history about the last actions that were performed. Similar to the suggestions in [9] it should be possible to let the system repeat the last dialog turn, e.g. by stating "Please repeat that". Also, the system should provide repair dialogs, e.g. if the system falsely turned on the light in the kitchen instead of the light in the living room. These repair dialogs should be as short as possible.

We analyzed these assumptions in a Wizard-of-Oz study that is described in the next section.

4. WIZARD-OF-OZ STUDY

We designed the Wizard-of-Oz study to have one participant solve an exercise of several tasks in a smart environment with just a small number of devices to control. The choice for a Wizard-of-Oz study lies within the potential to reproduce the same kind of errors on the same dialog step.

We developed a mock-up implementation that simulates the behavior of a small system containing the control of light and shutter in one room, namely a living room with a door to the terrace.

This program consists of two windows, one for the participant and one for the experimenter. The window for the experimenter (see Figure 1) shows a selection of possible answers to the instructions or answers given by the study participators in the context of the dialog flow.

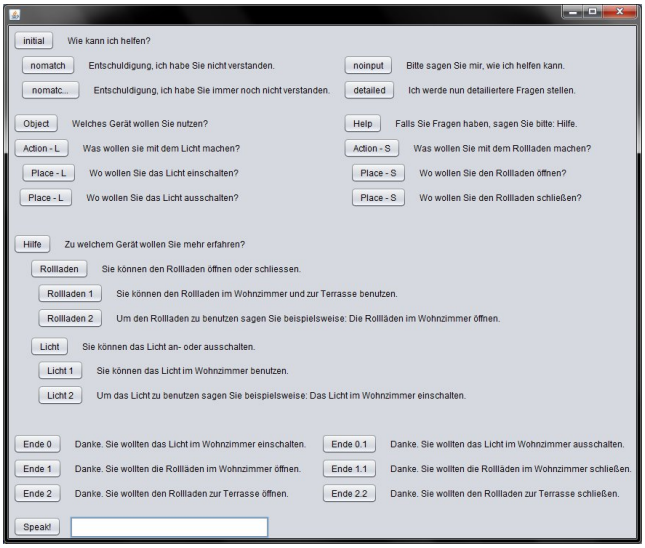

Figure 1: View of the experimenters control window

The possibilities to react to the participators input are structured with regard to the dialog flow from top to bottom with an initial statement and their possible outputs in erroneous situations. The next block contains the answers for partially understood statements and the hint for using the help, followed by a block for the actual assistive help. The last section contains the closure of the dialog and a text field to give the experimenter the possibility to write down an individual answer if needed.

There were 10 participants for this study separated into two groups: Experts and uninitiated users. Both groups had participants that did not have any experiences with voice controlled applications, except for telephony systems like cinema or banking systems and others who have tried systems like Siri and used them on a regular basis.

Experts were considered to have a deeper knowledge in either computer science, voice based interaction, human computer interaction, linguistics or psychology. The uninitiated group consisted of students of different majors.

There were five tasks for each participant to solve during this study. The first one served as an introduction to the system. Here, the participants were asked to turn on the light.

The evaluation took place in a quiet room providing a screen for the participants showing her view of the demo application with a living room environment that behaves according to the dialog (see Figure 2). Depending on the user's command the experimenter could e.g. exchange the image by another one with the same scenario but all shutters

Figure 2: Living room with shutters opened and light off

closed and the light turned on (see Figure 2). Another set-

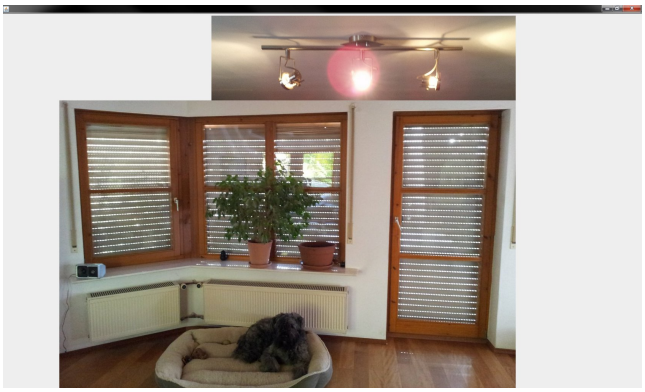

Figure 3: Living room with shutters closed and light on

ting with the shutters partially closed and the light turned on is shown in Figure 4. The study started with the latter

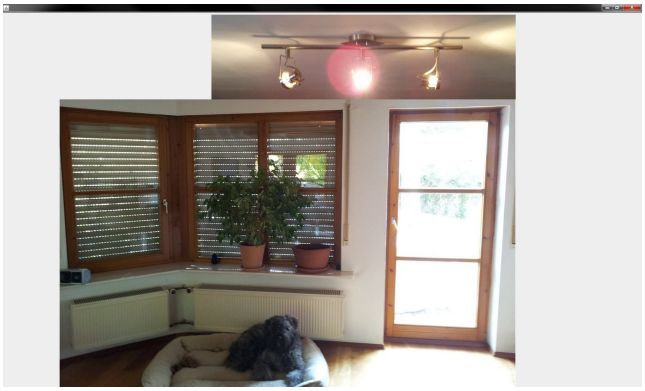

Figure 4: Living room with shutters partially closed and light on

scenario and the participants were asked to open the shutter. No errors were destined to occur in this task to strengthen the trust of the participant in this system.

The participant then should close the shutter. In this case partial recognition was simulated by rejecting the verb in the participant's command.

The third task was to turn off the light and was designed to recognize nothing on the first two inputs.

As the fourth task, the participant was asked to open the shutter to the garden but the word *garden* was unknown to the system. This task should encourage the usage of help and reaction to implicit answers due to partial recognition.

In the last task the participant should open the remaining shutter causing a *no input* error and to turn off the light without causing an error to not leave the participant frustrated of causing too much errors.

Afterwards, a short discussion was held to learn about the impressions of the participant while executing the tasks. Important issues during the discussion have been the intuitiveness of the solutions, the help and formulations of the system and its consistency and the feeling of being in control regarding the interaction.

Each evaluation took about 15 minutes and each participant was evaluated separately. The participators were neither told that the application was a mock-up and the speech recognition was manually performed by a person nor further information other than the tasks were given to not alter the behavior while solving the tasks.

5. SUMMARY OF OBSERVATIONS

The first task was solved as intended by three participators, the rest did not use the explicitly mentioned location *living room* and triggered the reaction *"Where do you want to turn on the light?"*.

The second task was similar to the first for those participants that didn't mention the location. It was also solved very fast by everyone.

Some participants first showed a noticeably difference in the pronunciation between talking to the experimenter and talking to the system. In this case the formulation was more carefully chosen and the pronunciation was clearer and slower. This changed in the first two tasks and became more natural.

As third task the participant should turn off the light in the living room. In the first two approaches nothing was recognized, so the system answered with an answer to the Level 1 error *"Sorry, I did not understand."* After the second output all participators selected their words more carefully and the language shifted from naturally to a short and more command like style. This effect was noticeable the most with the experts. No one formulated the complete sentence they used at first but just answered the questions formulated by the system.

The task designed to be the most difficult one was the fourth. The participant were asked to open the shutter to the garden but the system only knew the word *terrace*. The way to operate the system which the the participants learned so far did not work any further. The error could not be solved in an implicit answer and question manner. The shortest way to solve this task was to ask where the user could perform the desired action.

Only five out of ten participants used help from the system to solve the task. Four participants solved this task by guessing the correct location and one participant could find no solution at all.

The participators who were not used to work with computer systems appeared to be the most flexible and therefore

faster than most participators with extensive knowledge of computer systems. Only one of the experts solved the task faster.

The *no input error* of the last task was solved in a confident manner by everyone using natural language.

In the discussion nearly every participant said that the interaction felt to be intuitive in solving the given problem. Everyone had the impression that the program was consistent in its behavior and nothing unforeseeable happened.

Positively mentioned were the implicit answers when the program reused the participant's wording or recognized the context. Some participants mentioned that this felt like a natural dialog between conversational partners.

Criticism was expressed on the fourth task since some participants felt overstrained because of the lacking knowledge in using the program. Another point of criticism was the small vocabulary that was intentionally used to provoke errors.

The participants that did not use any help from the system in the fourth task said that they did not even thought of using any help in that situation because they normally wouldn't use the help function in other programs. But everyone thought it would be appealing if it were possible to formulate a question freely to a computer system by voice.

Two experts criticized that the program provided acoustical feedback in an affirmative sentence. If the result of a command was visible, the program should not formulate another sentence.

6. DISCUSSION

Although a population of only ten participants is not significant, the evaluation showed some interesting trends.

The most striking result was that the tasks were solved best by those persons with the least technological background. It appears as if those persons whose mental model [7] was not heavily influenced by technological background knowledge had less difficulties in interacting with the system. For them, the interaction was judged to be more appealing. This finding could be exploited e.g. in AAL scenarios where elderly persons have to operate a system. Usually, they are not very familiar with technology. However, this may require a dedicated study.

Overall the system was judged positive. Frustration was observed only while conducting the fourth task which was difficult to solve by design. The participants said that it was fun to operate the system and that they could imagine to use such a system in their homes if it were available.

While conducting the experiment it occurred that the system's responses were lagging. This was caused by the necessity to move the mouse at the experimenter's screen to the desired position. However, a lag of 4-6 seconds seemed to have no negative impact onto the dialog flow.

Especially the use of the patterns Implicit Confirmation, Selection From A List and Three Tiered Confidence was judged positive. The participants had the impression to have a dialog with the system. In contrast, the explicitly offered help functionality was not perceived as a dialog and was not considered to be natural.

Similarly, patterns like Explicit Confirmation or Global Error Correction were more of a kind that increased the feeling of not being in control. In these cases the participants did not perceive the interaction as a dialog.

7. CONCLUSION & OUTLOOK

In this paper we analyzed the potential of a mixed-initiative approach to deal with error situations when controlling devices in smart home environments. Therefore, we employed patterns known from the design of voice user interfaces in telephony environments and discussed how they could be used to handle these situations.

We conducted a wizard-of-oz study where we tested the usage of different error management patterns in erroneous situations.

The patterns Implicit Confirmation, Selection From A List and Three Tiered Confidence were good candidates to be used in controlling smart environments. They gave the users the feeling of being in control of the system. Moreover, they were suitable means to let the user perceive the overall interaction as a dialog, especially when they had only little technological background. The patterns Explicit Confirmation or Global Error Correction were not very suitable in this context.

In contrast to the assumption that command & control settings are more of a kind of fire and forget, we observed a dialog conducted between the user and the system which increased the user experience. With a conversational approach users still felt in control although they realized that the system did not always understand what they said.

As a next step we plan to use our results to develop a system in a real environment to verify the trends that we observed.

8. REFERENCES

[1] M. H. Cohen, M. H. Cohen, J. P. Giangola, and J. Balogh. *Voice user interface design.* Addison-Wesley, 2004.

[2] D. Duff, B. Gates, and S. LuperFoy. An architecture for spoken dialogue management. In *Proceedings of the 1996 International Conference on Speech and Language Processing (ICSLP)*, 1996.

[3] L. Dybkjær and N. O. Bernsen. Usability issues in spoken dialogue systems. *Natural Language Engineering*, 6(3&4):243–271, 2000.

[4] T. Heinroth, M. Grotz, F. Nothdurft, and W. Minker. Adaptive speech understanding for intuitive model-based spoken dialogues. In *Proc. LREC*, pages 1281–1288, 2012.

[5] T. Heinroth and W. Minker. *The OwlSpeak Adaptive Spoken Dialogue Manager*, chapter 4, pages 65–111. Springer, 2013.

[6] G. Meszaros and J. Doble. Pattern languages of program design 3. chapter A pattern language for pattern writing, pages 529–574. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[7] D. A. Norman. *The design of everyday things.* Basic Books (AZ), 2002.

[8] A. Oulasvirta, K.-P. Engelbrecht, A. Jameson, and S. Möller. Communication failures in the speech-based control of smart home systems. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 135–143. IET, 2007.

[9] H. Sagawa, T. Mitaruma, and E. Nyberg. Correction grammars for error handling in a speech dialog system. In *Proceedings of HLT-NAACL 2004: Short*

Papers, pages 61–64, 2004.

[10] D. Schnelle-Walka. A pattern language for error management in voice user interfaces. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, page 8. ACM, 2010.

[11] D. Schnelle-Walka, J. Arndt, and S. Feldes. Towards mixed-initiave concepts in smart environments. In *Proceedings of Workshop Interacting with Smart Objects*, Feb 2011.

[12] M. Turunen and J. Hakulinen. Agent-based error handling in spoken dialogue systems. In *Proceedings of Eurospeech*, volume 2001, pages 2189–2192, 2001.