Supporting the Wizard: Interface Improvements in Wizard of Oz Studies

Stephan Schlögl
Trinity College Dublin
Ireland
schlogls@scss.tcd.ie

Anne Schneider
Trinity College Dublin
Ireland
schneia@scss.tcd.ie

Saturnino Luz
Trinity College Dublin
Ireland
Saturnino Luz@scss.tcd.ie

Gavin Doherty
Trinity College Dublin
Ireland
Gavin.Doherty@scss.tcd.ie

Prototyping early in the design process is important for the development of high-quality software. Sketches and wireframes are effective artefacts that inform the design of applications based on Graphical User Interfaces. For applications using speech and Language Technologies (LTC) the Wizard of Oz method aims to fullfil this task. In order to support the demanding task of the wizard, however an optimal wizard interface is desirable. While several wizard interfaces have been built to date, most of them were designed for designated experiments. The possibilities of a generic wizard interface that would address the difficulties of the wizard task across the boundaries of varying experiment settings have remained largely unexplored. In this paper we report on two experiments that aimed at exploring the wizard task in order to inform the design of a univerals wizard interface for testing LTCs.

Wizard of Oz; User Evaluation; Language Technology; Prototyping

1. INTRODUCTION

Early and iterative prototyping is crucial for building high quality products. Low-fidelity prototyping methods like sketching and wire-framing have become standard in software development. Applications using Language Technologies like Automatic Speech Recognition, Speech Synthesis and Machine Translation have increased as their quality has significantly improved over the last years. Yet, processes and tools that allow for an early stage evaluation of those applications are still scarce. One method that is used to prototype language-based software is the Wizard of Oz (WOZ) technique (cf. Gould et al. (1983)). WOZ uses a human wizard to mimic some of the functionality of a system. This allows to evaluate possible user experiences without the need for a fully functional real-world product. Often it is used to test Language Technology Components (LTC), to inform the design of the human-machine dialogue, or to collect language corpora. However, as Salber and Coutaz (1993) point out, the human wizard needs to deal with a highly demanding cognitive task.

In this paper we report on work in progress, which aims at understanding the task of the wizard and tries to identify ways to support him/her through an improved software interface. Therefore a study was conducted in the context of a specific scenario where a system recommends appropriate Internet broadband connections to German speaking customers. The scenario and the developed prototypes were employed in a experiment series that investigated the output of machine translation systems and the combination of machine translation and speech synthesis in interactive applications.

The results we present in this paper focus on finding an optimal wizard interface. As such our experiments highlight two major issues a WOZ interface is confronted with, namely: striking the right balance between automating the wizard's task while keeping the wizard in control of the interaction, and providing appropriate mechanisms to deal with response timing issues from the wizard's as well as the user's perspective. We will use this insight to inform the design of an integrated WOZ prototyping framework focusing on speech and language-based interaction.

2. MOTIVATION

As language technology improved, LTCs have started to be used more frequently in systems of our everyday lives. Speech-based navigation in cars keep a driver's attention on the road, and translation tools such as GOOGLE TRANSLATE and YAHOO! BABEL FISH help us to understand text written in a foreign language. Call centres use Interactive

Voice Response Systems to handle their workload and reduce costs, and Emotional Conversational Agents try to augment our learning experiences. Still, due to the relatively novel nature of some of those applications it is important that software using LTCs is tested early in the design process. Unlike prototyping that assesses applications which are mainly based on Graphical User Interfaces (GUI), obtaining early feedback for speech- and language-based software can be both cost and time intensive. For those cases the Wizard of Oz method has been found to be an appropriate evaluation method (Buxton (2007)). Yet, a comparably high amount of development effort is necessary to create a wizard interface. In addition, unlike most prototyping methods for GUI-based applications, WOZ depends on the actions of a human at runtime. Whereas with GUI commands a certain behaviour can be defined in advance by referring to concrete events like mouse-clicks and keyboard-entries, in speech-based interaction such bindings are less direct. In the case of WOZ this interpretation is done by a wizard, which opens up possibilities for inconsistent timing and behaviour that could undermine the results of a study. Another goal is to make sure that the human intervention is imperceptible to a test user interacting with the system. One way to deal with these challenges is to provide the wizard with an appropriate interface that helps to fulfil his/her task efficiently.

In the following we discuss first insights into the problems a wizard faces, and how certain design choices might help to solve them. We begin with a study of the literature that reports on the usage of WOZ, which is followed by an illustration of the underlying scenario we used for our study. Subsequently, we describe two prototypes to evaluate different design ideas. Two small-scale experiments aimed at understanding the wizard's task, and at investigating the different aspects of the wizard interface. We report on the outcome of those experiments and discuss how our findings influence the design process of a future wizard interface.

3. RELATED WORK

Wizard of Oz has been used as a design method for more than 40 years. Erdman and Neal (1971) employed it to test the concept of a self-service airline ticket kiosk and Gould et al. (1983) evaluated the benefits of 'The Listening Typewriter'. The method primarily serves the purpose of informing the design of a not yet existing technology component. As such Dahlbäck et al. (1993) highlighted its use as an interaction design technique and other researcher stressed its application by pointing to the importance of early user studies informing the design and development of computer systems (e.g. Gould and Lewis (1985), Buxton (2007)).

WOZ studies have been found to be an efficient way of testing LTCs and software using LTCs. Its application supports the development of rich dialogue models and allows the designer to produce a more natural interaction. Recent utilisations of the WOZ method include among others the testing of multi-modal information retrieval (Rajman et al. (2006)), the development of a spoken dialogue system (Karpov et al. (2008)) and the simulation of a virtual doorman (Mäkelä et al. (2001)). In a more open domain Bradley et al. (2009) applied WOZ to inform the design of a web-based companion.

Despite the widespread use of the WOZ method, most of the wizard interfaces employed in the above mentioned studies were built for one set of experiments only. We are not aware of research efforts devoted to understanding the role of the wizard. As such we believe that there is a lack of design recommendations for interfaces that cater for the needs of a wizard, as a user of a WOZ tool.

4. THE UNDERLYING SCENARIO

The WOZ prototyping framework being developed in our group finds application in an experiment series, which aims at investigating the output of machine translation systems and the combination of machine translation and speech synthesis in interactive applications. For this purpose an underlying scenario simulates the interaction between a German speaking customer searching for an Internet product (broadband connection) and a computer system recommending appropriate bundles based on the customer's needs. The system is supposed to understand spoken input and uses text or speech to output questions and recommendations to the customer. As an additional feature, the customer is told, that the interaction can take place in German. To support this the system would use a built-in machine translation mechanism.

In order to define the possible system utterances for this interaction we looked at the domain data and the possible interaction strategies and scripted a preliminary dialogue. This resulted in 32 utterances (10 of which had open slots for context specific information) and a list of slot fillers. The designed dialogue was then tested for accuracy and completeness using a chat tool and applying simple copy and paste mechanisms. To support the interaction in German, the different dialogue utterances were subsequently translated using a machine translation system.

5. FIRST PROTOTYPE

In accordance to this pre-defined dialogue an interface was built, which allows a human (the wizard) to simulate the main functionalities of the intended computer system (i.e. natural language understanding, dialogue management and natural language generation).

The interface of the prototype (see Figure 1) was split into two areas. The left side with a grey background was dedicated to the dialogue. The right side contained the result set comprising the concrete offers for Internet bundles.

Our intention was to support the wizard by unburdening his work as much as possible. Therefore, we decided to let the system guide through the dialogue progress by automatically adjusting possible responses represented in a green box on the left side under the label 'Respond'. When the wizard chose an utterance that would lead to the next stage in the dialogue flow the green box was updated automatically showing a new set of appropriate responses. The wizard could however switch between dialogue stages also manually by clicking on the links in the yellow 'Dialogue flow' box. Furthermore the amount of possible recommendations on the right hand side was decreased automatically, depending on the use of certain utterances within the dialogue. In the dialogue literature this process is referred to as slot filling (cf. McTear (2002)). The wizard could in addition apply filters to further reduce the set of recommendations.

Utterances that were aimed at helping the wizard recover from misunderstandings, could be chosen from an orange box in the 'Not understood' section of the screen. Since this set of responses was useful for the wizard at any time during the interaction it did not change and was displayed constantly.

5.1. Evaluation of the first Prototype

In order to obtain a general understanding of the task of the wizard a first experiment was conducted (cf. Schlögl et al. (2010)) using the described wizard interface prototype. An informal, qualitative usability test approach (cf. Nielsen (1993)) was chosen with four participants acting as wizards. Two members of our research team as well as two externals were asked to take part. They had different familiarity with the WOZ method in general and the experiment setting in particular. One person, who was familiar with the WOZ technique, was the designer of the dialogue and therefore knew about the dialogue structure and its utterances. Another person was acquainted with the WOZ method but not involved in the experiment design. The two remaining participants had no previous experience with WOZ and were not involved in the experiment design. Participation in the experiment was voluntary and not rewarded.

The participants received a one-page description of the task and were allowed to explore the wizard interface for about five minutes. Actions of the wizards were logged, and we used the SCREENFLOW screen casting software to capture the wizards' screen. In addition a person was observing them and taking notes. Using headphones ensured that the wizards were masked from environmental noise. A retrospective analysis was conducted after the task in order to shed light on some of their actions at runtime.

The developer of the wizard interface acted as customer. He was sitting in a different room looking at a web-based client interface running on a 15-inch computer screen. His main mode for interacting with the wizards was through a microphone. SKYPE was used to transmit the speech.

In a supplementary independent trial the developer of the interface acted as the wizard for ten experiment runs with real test users. These interactions were also recorded and in addition he kept a protocol of his problems using the WOZ interface.

5.2. Results

This preliminary study gave insight into several problems a wizard is facing when running an experiment. We identified three major issues. The most fundamental of which was that wizards had difficulties to follow and control the dialogue flow, which led to problems of finding the right response utterances. This behaviour applied not only to novice wizards but also to the more experienced ones who were familiar with the dialogue and the wizard interface. One reason for this was the layout of the interface into logical areas, which was not integrated enough and therefore lacked intuitiveness. Wizards where not able to predict where they would find the next utterance, especially when they were supposed to switch between two different areas. Consequently they expressed a desire to be able to notify the customer that they were processing information and needed more time.

Secondly we were able to identify a functional problem of the interface leading to difficulties with the dialogue flow. The participants were confused by the filters that enabled them to sort the utterances recommending Internet connection bundles according to certain criteria (e.g. price,

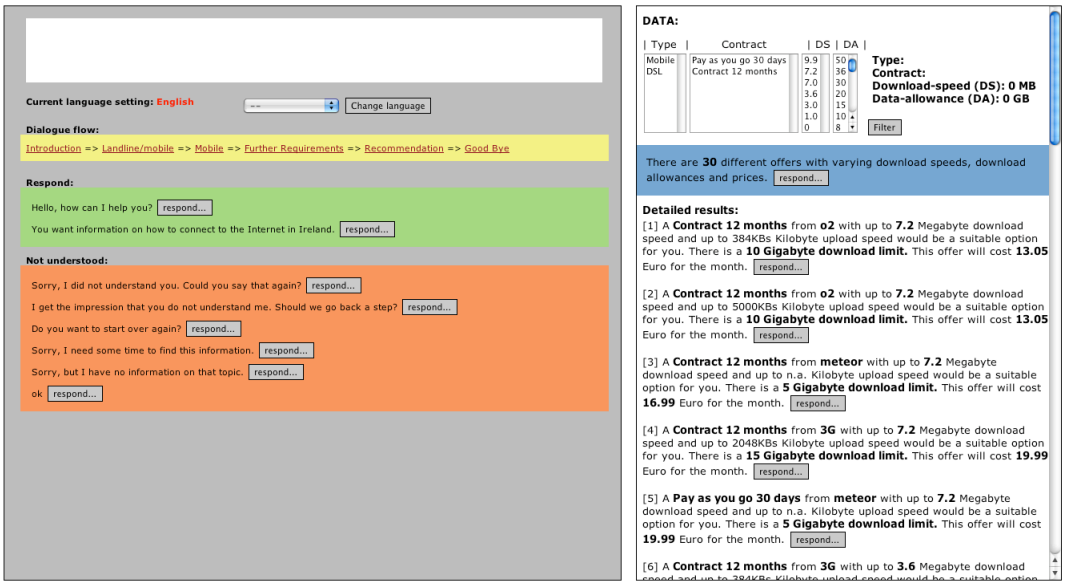

Figure 1: *First prototype of the wizard interface.*

bandwidth). The adjustment of possible offers was usually done automatically depending on the dialogue flow, however, it was repeatedly observed that the participants tried to manually adjust the filters even though there was no need for such behaviour. In the interviews, participants explained that the functionality of this slot-filling mechanism was not transparent enough. This demonstrated that even though any functionality that might facilitate the highly demanding cognitive task of the wizard by automating parts of it would appear to be helpful, wizards still need to be in full control of the dialogue. If supportive functionality is implemented, this functionality needs to be transparent.

Finally, from a dialogue perspective certain timing issues were identified. Due to the text-based output of the simulated system, a wizard could not be sure how long a customer would need to read and understand an utterance. For long utterances this sometimes outlasted a ten second threshold. In this case a wizard would send an additional utterance (e.g. 'I can't understand you') in order to check on a customer's status. This behaviour was observed with all wizards and seemed consistent. Another timing issue seemed salient in some cases where a wizard would use two subsequent utterances without waiting for a response from the test subject. Even though one could argue that such behaviour should be prevented by the wizard interface so as to reflect more realistically the capabilities of a real computer system, it also highlights the qualities of the Wizard of Oz method for identifying flaws and inconsistencies in the dialogue design. In this case the two subsequent utterances should have been simply combined.

6. SECOND PROTOTYPE

Based on the results from the first experiment a new prototype was built (see Figure 2). The prototype tried to address the identified problems focussing on the three major issues: control over the dialogue, handling domain data, and timing.

With the intention to support the wizard navigate the dialogue by making the different dialogue steps apparent, a tabbed dialogue structure was introduced. In addition a dedicated area for frequently used utterances was added to the right side of the interface. Utterances that could not be assigned to a single dialogue step were either added multiple times or displayed in the separate area on the screen dedicated to the frequently used utterances.

The second major change was focusing on the domain data (i.e. utterances recommending Internet bundles) and how it is filtered. The slot-filling mechanism, that was used in the first study and confused most wizards, was replaced by a simpler version using radio buttons, where the wizards were responsible for handling this task manually.

Finally to address some of the identified timing issues a basic notification mechanism was integrated, that should serve as a link between the wizard and the customer. The goal was to inform a wizard about a test users status. Therefore an information panel was placed in the head of the interface consisting of four possible user states: 'No active session' (displayed in black), 'Session started' (displayed in red), 'Session running' (displayed in green), and 'Session stopped' (displayed in blue). As soon as the

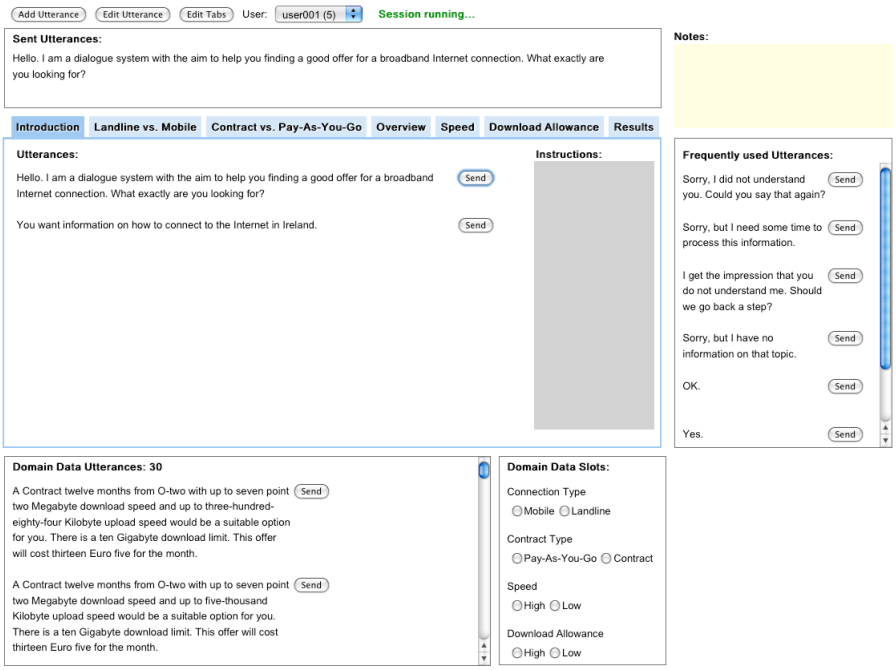

Figure 2: *Second prototype of the wizard interface using a tabbed dialogue structure, a simplified slot-filling mechanism and notifications.*

customer started the session the wizard was notified by the information state switching from black ('No active Session') to red ('Session started'). When the wizard replied the text switched to green ('Session running'). Finally the customer changed the status to blue ('Session stopped') by clicking a button at the end of an experiment. Using different colours for the different states was used to catch the wizard's attention.

In preparation for future experiments we also added some basic editing functions to the wizard interface. Those functions allow to add, delete and change utterances as well as tabs. Furthermore it is now possible to rearrange the order of utterances and move them to different tabs. A section for notes as well as instruction panels added to every tab were meant to help wizards to be more consistent. However, this final set of features will receive a separate round of testing and was not part of the here described evaluation.

6.1. Evaluation of the second Prototype

This follow-up experiment aimed at testing some of the concepts identified in the first experiment by using the new wizard interface. This time three participants acted as the wizard all of whom were from outside our research team. The set-up resembled the preliminary experiment, except that the designer of the dialogue was acting as a customer and the developer of the wizard interface was observing the wizards. None of the participants was familiar with the WOZ method even though two of them can be classified as potential users of a WOZ tool since they develop software for natural language processing applications. Participants were introduced to the WOZ method in general and their task in this specific experiment. Thereafter they were given as much time as they needed to explore the interface, read the utterances and ask questions, before the experiment started.

6.2. Results

The results of this second experiment showed an improvement for the revised prototype over the initial one. Looking at how the participants navigated through the dialogue the new interface layout using tabs seemed promising. Wizards had no problems moving between the dialogue stages in time.

Also the handling of the domain data (slot-filling) seemed to be easier for the wizards using the radio button layout. All of them used the radio buttons even though sometimes they forgot to do so immediately after receiving the relevant information from the customer. Therefore they needed to adapt them right before sending recommendation utterances. When asking them whether they had preferred the slot-filling to be done automatically all three of them expressed their preference for the manual approach.

However, it is still an open issue how to support the wizard in estimating the time a customer needs to read an utterance. The notification mechanism that was implemented in order to tackle this problem was

helpful but it did not go far enough. A precise status information on the client's reading progress would be desireable. Furthermore, the notification mechanism is only targeting the wizard. For the future, however, it is planned to integrate a similar system on the client site (i.e. for the customer) to automatically give feedback that the system is processing the data.

In summary it can be said that the new wizard interface showed an improvement in terms of controlling the dialogue and filtering domain data. However, further adjustments are necessary. Supporting the wizard in estimating a test user's reading and processing time of text utterances remains an open issue.

7. CONCLUSION AND FUTURE WORK

We have presented work in progress that aims at understanding the task of the wizard when running WOZ experiments. This research aims to inform the design of a more supportive wizard interface to be implemented in an integrated WOZ prototyping framework. Results of two small experiments were discussed which show how certain design choices can help to disburden the task of the wizard.

Future work will look at the performance of wizards over time and how this might lead to additional requirements with regards to customizing and adjusting a wizard interface. In addition we plan to evaluate some features that have not undergone any experimentation so far (e.g. notes and instructions). Finally, more effort needs to go into investing how wizards best handle domain data. In the presented experiments this data was relatively small and easy to navigate. Further studies are required to explore how more complex data structures influence the wizard's workload and how this impacts the outcome of a WOZ experiment.

8. ACKNOWLEDGEMENTS

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

9. REFERENCES

Bradley, J., Mival, O. and Benyon, D. (2009). Wizard of Oz Experiments for Companions. *Proceedings of BCS HCI*, 115-119.

Buxton, B. (2007). *Sketching User Experiences.* Morgan Kaufman, Oxford, UK.

Dahlbäck, N., Jönsson, A. and Ahrenberg, A. (1993). Wizard of Oz Studies: Why and How. *Human Factors 13*, 521-531.

Erdman, R. L. and Neal, A. S. (1971). Laboratory vs. Field Experimentation in Human Factors: An Evaluation of an Experimental Self-service Airline Ticket Vendor. *Proceedings of IUI*, 193-200.

Gould, J. D., Conti, J. and Hovanvecz, T. (1983). Composing Letters with a Simulated Listening Typewriter. *Communications of the ACM 26*(4), 295-308.

Gould, J. D. and Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of ACM 28*(3), 300-311.

Karpov, A., Ronzhin, A. and Leontyeva, A. (2008). A Semi-automatic Wizard of Oz Technique for Let's Fly Spoken Dialogue System. *Lecture Notes in Computer Science: Text, Speech and Dialogue, 5246*. 585-592.

Mäkelä, K., Salonen, E.-P., Turunen, M., Hakulinen, J. and Raisamo, R. (2001). Conducting a Wizard of Oz Experiment on a Ubiquitous Computing System Doorman. *Proceedings of IPNMD Workshop*, 115-119.

McTear, M. F. (2002). Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Comput. Surv. 34*(1), 90-169.

Nielsen, J. (1993). *Usability Engineering*. Academic Press, Boston, MA.

Ogden, W. C., Bernick, P. and Helander, M. (eds) (1988). *Handbook of Human-Computer Interaction.* Using Natural Language Interfaces. Elsevier Science Publishers B. V., Amsterdam, The Netherlands.

Rajman, M., Ailomaa, M., Lisowska, A., Melchiar, M. and Armstrong, S. (2006). Extending the Wizard of Oz Methodology for Language-enabled Multimodal Systems. *Proceedings of LREC*, 2531-2536.

Salber, D. and Coutaz, J. (1993). A Wizard of Oz Platform for the Study of Multimodal Systems. *Proceedings of INTERACT and CHI*, 95-96.

Schlögl, S., Doherty, G., Karamanis, N., Schneider, A. and Luz, S. (2010). Observing the Wizard: In Search of a Generic Interface for Wizard of Oz Studies. *Proceedings of iHCI*, 43-50.