



**HAL**  
open science

# Sketching Language: User-Centered Design of a Wizard of Oz Prototyping Framework

Stephan Schlögl

► **To cite this version:**

Stephan Schlögl. Sketching Language: User-Centered Design of a Wizard of Oz Prototyping Framework. 13th International Conference on Human-Computer Interaction (INTERACT), Sep 2011, Lisbon, Portugal. pp.422-425, 10.1007/978-3-642-23768-3\_48 . hal-01596893

**HAL Id: hal-01596893**

**<https://inria.hal.science/hal-01596893v1>**

Submitted on 28 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Sketching Language: User-Centered Design of a Wizard of Oz Prototyping Framework

Stephan Schlögl

School of Computer Science & Statistics  
Trinity College, University of Dublin  
O'Reilly Institute, Dublin 2, Ireland  
schlogls@scss.tcd.ie

## Supervisor

Dr. Gavin Doherty  
Gavin.Doherty@scss.tcd.ie

## Research Area

Wizard of Oz prototyping with modern Language Technology Components;

## Research Topic

This research aims at understanding the challenges involved in running Wizard of OZ experiments and searches for an optimal software interface supporting the actions of the wizard.

## Research Description

Combining novel speech and language technologies into systems across diverse domains poses significant software engineering challenges. Assuring the usability of the resulting applications is an important, and somewhat under-researched, issue in this area. The use of Language Technology Components (LTC), i.e. Automatic Speech Recognition, Machine Translation and Speech Synthesis, has significantly increased in recent years as their performance has improved. Examples include speech-based interaction in cars to keep a driver's attention on the road and the use of web-based translation tools such as *Google Translate* and *Yahoo! Babel fish*, to understand text that is written in a foreign language.

Due to the fallible nature of the technology involved in such applications, one must ensure that usability and human factors are adequately accounted for. As with applications based on a Graphical User Interface (GUI) software using LTCs needs to be tested early in the design process. While low-fidelity prototypes assessing GUI applications such as sketches and wireframes can be built relatively quick and

inexpensively, the development of prototypes evaluating the use of LTCs tends to be both cost and time intensive. An easier and more efficient way of creating those prototypes is desirable. One technique that has been used to test applications involving LTCs is Wizard of Oz (WOZ). In a WOZ experiment a human wizard mimics some of the functionality of a system. Since the technical requirements for such a prototype can be reduced to a minimum, this technique is particularly useful for early stage evaluations and therefore can be seen as a good candidate for addressing the lack of low-fidelity prototyping methods supporting the use of LTCs. An important difference between prototyping GUI based applications and conducting a WOZ experiment is, however, that the latter depends on the actions of the human wizard. Whereas with GUI commands a specific system behavior can be defined in advance by referring to concrete events like mouse-clicks and keyboard-entries, with language technology this clear binding is missing. Here the interaction rather depends on the way a user's input is interpreted. In a WOZ experiment the wizard is responsible for this highly demanding cognitive task [5]. Supporting this role with a usable wizard interface seems therefore crucial.

While several wizard interfaces have been built to date (e.g. [4], [2], [3] and [1]), most of them were designed with specific experiments in mind. The more general issue of understanding the task of the wizard, in order to develop a generic WOZ interface that can be used across different experiments and settings, has remained largely unexplored. The goal of this research is therefore to create a WOZ prototyping framework that aims at a template-based generation of wizard and client interfaces for different application scenarios. Besides providing an environment that offers an easy way of prototyping LTC-based software it is further an important goal of this work to understand and optimally support the task of the wizard.

### **Research Questions**

Based on the stated goals the following research questions arise:

- What is the design space for applications using LTCs and for which of those scenarios can WOZ realistically be used as a low-fidelity prototyping method?
- What are the various aspects of the wizard task and especially what kind of problems does a wizard need to deal with when designing and running an experiment?
- Can the function of the wizard be supported and if yes what features can be implemented into a WOZ prototyping framework that would disburden this task?

### **Research Hypothesis**

The main hypothesis for this research is that by providing an optimal interface for the wizard the complexity of the wizard task can be reduced and that this improvement would increase the wizard's overall performance in terms of consistency and machine-like behavior. It is furthermore assumed that it is possible to structure and classify WOZ experiments, and that this helps to build a software framework, which allows for a somewhat generic creation of wizard and client interfaces.

## Research Methodology

In order to fully understand and optimally support the design and conduct of WOZ experiments a User-Centered Design (UCD) methodology is followed. UCD encompasses the following steps that iteratively lead through the planning, design and development of a new product:

- Specify the context of use: This first stage of UCD focuses on identifying the people who will use the future product, the conditions under which it will be used and the reason why it is used. For the proposed research this contextual information will on the one hand be drawn from analyzing the literature, and on the other hand be obtained through talking to wizards and observing their actions.
- Specify requirements: Based on the context of use the second stage of UCD defines requirements that need to be met for the product to be successful. Requirements can be separated into business requirements e.g. the functionality of the product and user requirements e.g. the usability of the product. In the case of WOZ the basic requirements will be derived from existing WOZ tools. Additional aspects will materialize whilst talking to wizards and analyzing their demands.
- Create design solutions: After defining the product requirements, different possible design solutions should be implemented. With the WOZ framework it is planned to employ various prototyping methods that reach from basic sketches on paper to more elaborate functional designs based on web technologies.
- Evaluate designs: As with most product development processes the evaluation phase is crucial for the quality of the product. Testing the different design solutions with actual users ensures appropriateness and applicability of the design and fosters its usability. In order to evaluate the different aspects of the proposed WOZ framework small user studies with different wizards, as well as longer lasting studies that look at the performance of a single wizard over time, are planned.

## Research Plan

Up to this point all of the above mentioned steps have undergone at least one iteration. The context of use was explored through testing and comparing existing prototyping products. A literature review highlighted several application scenarios and helped to define concrete requirements for a WOZ prototyping framework. Finally, a first prototype was developed and evaluated. Valuable results were produced and published in two conference papers (cf. [6] and [7]). In order to further proceed with the research the following steps were identified:

- Run several WOZ experiments in different experimental settings in order to understand the task of the wizard and find common problem patterns.
- Address these problems and define a modular wizard interface that optimally supports the wizard and allows for the testing of different application scenarios.
- Find a configurable software architecture that on the one hand supports the creation of wizard and client interfaces and on the other hand allows for a flexible integration of LTCs.

- Design and implement a WOZ prototyping framework based on the defined software architecture and evaluate its quality of supporting the wizard task.
- Evaluate the process of creating wizard and client interfaces for specific application scenarios and the efficiency of preparing, conducting and analyzing WOZ experiments. Finally, evaluate the diversity of application scenarios that are supported i.e. the amount of scenarios for which the framework can be used.

### **Expected Research Contribution**

This research focuses on WOZ from the perspective of prototyping LTC-based software applications. As such it involves the analysis and comparison of existing prototyping tools, looks at their usability and assesses their support for different LTCs. A second aspect of the research looks more specifically at WOZ experiments in order to investigate the task of the wizard and the challenges a person in this role is facing. By doing so the goal is to identify supportive factors to be implemented in a novel WOZ framework that allows for prototyping, testing and evaluating software applications, which employ different kinds of LTCs.

In summary the overall contribution of the anticipated research outcome is to expand the body of knowledge on WOZ prototyping, specifically on the challenges of planning, designing and running WOZ studies, and to use this insight to create an application framework that offers an easy way of building WOZ prototypes that make use of different LTCs. Sub-contributions include the definition of a categorization model for WOZ experiments, the identification and classification of problems influencing a wizard's performance, and a software architecture that allows for the flexible combination of different LTCs.

### **References**

1. Bradley, J., Mival, O., Benyon, D.: Wizard of Oz Experiments for Companions. Proceedings of the BCS Conference on Human-Computer Interaction, 313–317 (2009)
2. Karpov, A., Ronzhin, A., Leontyeva, A.: A Semi-automatic Wizard of Oz Technique for Let's Fly Spoken Dialogue System. In: Sojka, P., Hork, A., Kopecek, I., Pala, K. (eds.) Text, Speech and Dialogue. LNCS, vol. 5246, 585–592. Springer, Heidelberg (2008)
3. Mäkelä, K., Salonen, E-P, Turunen, M., Hakulinen, J., Raisamo, R.: Conducting a Wizard of Oz Experiment on a Ubiquitous Computing System Doorman. Proceedings of the IPNMD Workshop, 115–119 (2001)
4. Rajman, M., Ailomaa, M., Lisowska, A., Melchiar, M., Armstrong, S.: Extending the Wizard of Oz Methodology for Language-enabled Multimodal Systems. Proceedings of LREC, 2531–2536 (2006)
5. Salber, D., Coutaz, J.: A Wizard of Oz Platform for the Study of Multimodal Systems. Proceedings of INTERACT and CHI, 95–96 (1993)
6. Schlögl, S., Doherty, G., Karamanis, N., Luz, S.: WebWOZ: A Wizard of Oz Prototyping Framework. Proceedings of EICS 2010, 109–114 (2010)
7. Schlögl, S., Doherty, G., Karamanis, N., Schneider, A., Luz, S.: Observing the Wizard: In Search of a Generic Interface for Wizard of Oz Studies. Proceedings of Irish HCI, 43–50 (2010)