A Wizard-of-Oz Setting for Multimodal Interaction

An Approach to User-Based Elicitation of Design Patterns*

Andreas Ratzka

Universität Regensburg
Institut für Medien-, Informations- und Kulturwissenschaft
93040 Regensburg
Andreas.Ratzka@sprachlit.uni-regensburg.de

Abstract
This paper describes a Wizard of Oz setting for multimodal interaction. The design of this experimental setup is one working package in our research on a design methodology for multimodal interaction based on design reuse and user participation. Design reuse can be accomplished by design pattern languages, which are a powerful means of information management for both system developers and usability experts. User involvement is performed via Wizard-of-Oz tests, which simulate system behaviour by means of human agents. The test scenario is managing e-mail with multimodal mobile and desktop devices. Our goal is to explore hypotheses on mult-
timodal interaction and thus identify pattern candidates for the design of multimo-
dal interaction.

1 Introduction

This paper describes the research design for Wizard of Oz simulations of multimo-
dal interactive systems and is situated in the context of methodological research on
multimodal interaction design. After a thorough review of literature on model-
based user interface design (such as Bürgy 2002, Calvary et al. 2003, da Silva 2001,
for multimodal interaction have been presented in previous work (Ratzka 2006,
Ratzka & Wolff 2006):

- Cross-application-aspects
- Workflow and task
- Contextual aspects, i.e. user, situation, environment, device
- Adaptation concept
- Application-specific aspects
- Interaction
- Presentation
- Software Architecture

Our methodology is based on design reuse and user participation. Design reuse can
be accomplished with so called design patterns and pattern languages. User partici-
pation and early user involvement is done with Wizard of Oz tests.

The term design pattern was firstly used by Alexander et al. (1977) in the domain of
architecture and later on introduced into the fields of software engineering (Gamma
et al. 1995, Trowbridge & Cunningham 2001) and user interface design (Borchers
2001, van Welie & van der Weer 2003). Design patterns describe solutions to well
known design problems in a structured and retrievable way. They relate together a
striking name, context and problem descriptions, the description of a rather abstract
solution, concrete examples, as well as strengths and pitfalls of this solution.

Patterns constitute an information management tool for system and interaction
design, especially when they are grouped together as interlinked pattern language,
which can be easily explored by the system designer. Furthermore, by means of pat-
tern languages, common design problems and their solutions can be called by their
name and can be discussed more easily by both software and usability experts.

Some of the high level research results gained in these domains, such as the often cited buzzwords *mutual disambiguation* and *redundant information display*, can be formulated as reusable solutions to recurring design problems and thus be described in a pattern format. In previous work, we have identified a still incomplete collection of patterns and pattern candidates for multimodal interaction from literature review (cf. Ratzka & Wolff 2006). The pattern *human action source distribution* will serve as an illustrative example:

### 1.1 Human Action Source Distribution

**Context** The input of different data types (common in image editing systems, such as selecting a tool from a palette and drawing a figure) sometimes requires a repetitive repositioning of the user’s hands between mouse and keyboard or a repetitive repositioning of the mouse cursor.

**Problem** Repetitive repositioning is non-productive and slows down interaction.

**Solution** Partition alternating subtasks into different action channels. Make use of two-handed (Raisamo & Räihä 2000) or speech-enhanced multimodal interaction (Hiyoshi & Shimazu 1994, Milota 2004).

Design patterns relate together problems and *well proven* solutions. In contrast to WIMP\(^1\)-based interaction and web-based applications, multimodality has not yet reached a high degree of dissemination in consumer electronics. Only few design questions can be countered with well proven solutions from research. Other research results hold for specialized contexts only and cannot be generalised. Pattern candidates won from them are proven solutions only in that context. This imposes that user participation is needed to verify whether a certain pattern really fits into the target context, whether a pattern candidate can be generalised to a valid pattern. Furthermore multimodal interaction invites to new design ideas which seem plausible but are not proven empirically. That’s why our literature review has to be complemented by user involvement, performed by means of Wizard of Oz tests.

---

\(^1\) WIMP: **Windows**, **Icons**, **Menus**, **Pointing device**.
2 Experimental Setting

Wizard of Oz simulations are a well known approach of informal prototyping in HCI research, especially in spoken interaction and multimodal interaction (cf. Balbo et al. 1993, Salber & Coutaz 1993, Womser-Hacker 1993, Hitzenberger & Womser-Hacker 1995, Cenek et al. 2005, Klemmer et al. 2000). They tend to counter a common problem of human computer interaction researchers when exploring new interface styles: At early stages of interface design, necessary components for interpreting user input are not yet available. The requirements needed to implement them are still unknown, but can be elicited by early user participation accomplished by means of simulations.

Thus Wizard of Oz settings allow user participation at the very first stages of interface design. Alternative approaches of informal prototyping comprise paper mock-ups and storyboarding, which are both exploited in multimodal user interface design (Chandler et al. 2002, Sinha & Landay 2001, Sinha & Landay 2003). In contrast to those techniques, Wizard of Oz tests provide to test users a more realistic look and feel so that design alternatives can be discussed more deeply.

To perform Wizard tests, you need at least two connected computers, one for the user interface, for the test person to perform interactive tasks, and another one for the Wizard, the person who interprets user input and simulates system behaviour. In multimodal scenarios, multiple Wizards may be required, e.g. one for input and one for output or one for each input modality.

This work focuses on the multimodal enhancement of WIMP-based desktop and mobile interaction with spoken elements. Whereas the WIMP-based behaviour is implemented directly, spoken input and multimodal fusion have to be simulated. The wizard interprets user input and may change the data model of the application accordingly, so that the view of the user interface is updated. He can trigger pre-formulated spoken output, unless an automatic speech feedback strategy is acti-
vated. This requires an efficient wizard interface, which can be controlled in real
time via shortcut keys.

The platform simulates a mobile multimodal digital assistant (mobile setting) as
well as a multimodal desktop e-mail client (desktop setting). The test person has to
perform the task provided by the test scenario, i.e. to organise her incoming mail,
organize and delegate meetings etc.

This scenario has been chosen because e-mail is one of the most successful and wide-
Furthermore, other research (Bers et al. 1998, Comeford et al. 2001, Miyazaki
2002) indicates the plausibility of multimodal interaction for this application do-
main. In three consecutive surveys (cognitive walkthrough with usability experts,
qualitative user study, quantitative user study) hypotheses on multimodal interac-
tion will be examined. In these studies, well determined test scenarios (in the style
of mail baskets, which are popular exercises in assessment centres) will be combined
with free exploration of the user interface by the test person.

One central question to be answered via this method concerns the feedback strat-
egy, especially the interplay and balance between graphical and acoustic system
output. Below the two pattern candidates system initiated modality hint and speech-
enhanced display are shortly outlined as examples. The former attempts to counter
the conflict between data appropriate modality allocation (such as graphical display
of long lists) and interaction history appropriate modality allocation (answering to
speech commands via speech replies). The latter one picks up the potential of mul-
timodal interaction to amplify virtually the display of a small device via speech out-
put, so that cluttered screens and navigation complexity can be minimised.

2.1 System Initiated Modality Hint

**Context** Multimodality provides the possibility to select modalities flexibly ac-
cording to task and data properties, user preferences, and context factors.

**Problem** Sometimes data properties and context factors are in conflict, that means,
they encourage the selection of different modalities. Both system and user
may lack information for selecting the most feasible interaction style, so
that neither pure system initiative nor pure user initiative seem appropri-
ate for modality selection.

**Solution** Point out alternative interaction styles to the user via spoken hints. If the
user requests information via speech commands (“get messages from John
Long”) and the search result is more easily displayed graphically (as a list
of messages) a spoken modality hint (such as “look at the display or say ‘read out’”) would complement the visual output but preserve interaction flexibility.

### 2.2 Speech-enhanced Display

**Context** Some interactive systems support a lot of different tasks. During daily work it is necessary to switch between tasks to get the information needed to perform the main task. When a user wants to answer an e-mail he might need other data which can be found in previous correspondence.

**Problem** Small displays make such “multi-tasking” very difficult. Different task windows cannot be displayed all at once, whereas switching between several task screens imposes high navigational efforts, such that users might lose orientation.

**Solution** If the display is too small for simultaneous presentation of several task windows, consider partitioning different task displays to different output modalities. Users might listen to new mail or to e-mails of a specific thread while composing a message on the same topic and preserving the visual display.

Both pattern candidates seem plausible but still lack empirical evidence which can be obtained only by user participation. In contrast to full functioning prototypes, Wizard of Oz simulations can react more flexibly to incrementally revealed interaction requirements which are unknown in the very first phases of development.

### 3 Application Architecture

From the perspective of software architecture human computer interaction can be viewed as the creation, selection, manipulation, and destruction of interaction objects. Multimodal interaction imposes the need of specialized interaction objects for covering modality specific interaction possibilities. In order to maintain consistency and to enable synergistic and alternating modality usage, these specialized interaction objects have to be inter-connected through standardized interaction protocols.
That’s why the interaction objects of the (simulated) user interfaces are organized in a distributed multi-agent architecture\(^2\) making use of observer, adapter, proxy, and other design patterns. This architecture is inspired by the MVC-extensions of Visual Works Smalltalk (Krasner & Pope 1988, Lewis 1995 p. 103 ff.) on the one hand and PAC-Amadeus (Coutaz 1987, Coutaz 1994, Nigay & Coutaz 1995) on the other hand.

In MVC- and PAC-based architectures observer-objects (here: views) are wrapped around each functional core object (here: model) to get notified about relevant data changes, i.e. to maintain consistency between the functional core and the user interfaces.

\[\text{Model} \quad \text{Controller} \quad \text{View 1} \quad \text{View 2}\]

```
set(1)
update()
update()
get()
return 1
```

\[\text{Controller} \quad \text{Model} \quad \text{View 1} \quad \text{View 2}\]

```
set(1)
update()
update()
get()
return 1
```

**Fig. 2: The Observer Pattern in the MVC Paradigm**

Adapters simplify the complex interface of some functional core objects to fit to the MVC protocol.

The proxy-pattern is used in order to introduce one or more degrees of indirection between objects belonging to the functional core of the application and those belonging to the user interface component.

Indirection is required when several interaction objects are combined together, e.g. one for selecting an interaction object and another one for modifying the current selection. This combination of the observer and proxy patterns is also called *subject channel* (view Lewis 1995 p. 103 ff.).

---

\(^2\) In our case we mean reactive agents and not intelligent agents (cf. Coutaz 1994).
Further indirection is required to make interaction objects remote-controllable which is needed in a distributed Wizard of Oz scenario but also in real working systems, where the speech recognition component and the user interface are running on separate computers.

4 Technological Base

The Wizard of Oz platform is realized as a distributed system consisting of a pocket PC and a desktop PC (mobile setting) or of two desktop PCs (desktop setting) respectively. The communication between the distributed components is realized via socket connections through USB, Bluetooth or Ethernet.

The platform is implemented using *ewe* (cf. Brereton 2005), a Java-based toolkit providing libraries for programming highly portable java applications. The application classes can be run with an ordinary J2SE runtime environment or, and that is necessary for mobile applications, using the Ewe virtual machine.

Applications to be run with the Ewe virtual machine are restricted to the use of the Ewe API (including a subset of java.lang classes), and third party libraries, e.g. your own code.

Applications run with a J2SE runtime environment can combine both standard Java and Ewe libraries. The resulting applications are less portable, but this approach provides more flexibility for implementing the desktop components, i.e. the Wizard interface and the user interface for the desktop setting using JFC (cf. Sun Microsystems 2006), which are easier to handle and more thoroughly documented than the Ewe GUI library.
5 Conclusion and Future Work

This paper describes a Wizard of Oz setting in the context of methodological research on multimodal interaction for mobile and desktop devices. Early user involvement performed via wizard tests complements the reuse-based approach of using design patterns. This is important because multimodal interaction is still innovative and therefore lacks a large inventory of proven design solutions. User involvement is needed in design phases as early as possible, which encourages the application of Wizard of Oz simulations.

This research intends to examine design aspects concerning task, context of use, adaptation, interaction and presentation in more detail by means of Wizard of Oz simulations.

In the short run we will perform qualitative user tests in order to validate the general plausibility of design decisions motivated by our pattern candidates. These user tests will be complemented by open interviews and focus group discussions in order to elicit further pattern candidates.

In the long term evaluation tools, eye-tracking as additional interaction modality, as well as a speech engine to allow more scalable user interface simulations have to be integrated in order to allow the collection of reliable quantitative data.

As a by-product, architectural considerations relevant for the design of multimodal user interfaces, which become apparent when implementing Wizard of Oz prototypes, will be collected and further systematised.

6 References


