Message Oriented Middleware for Flexible Wizard of Oz Experiments in HCI

Mirko Otto, Rafael Friesen, and Dietmar Rösner

Department of Knowledge Processing and Language Engineering
Otto-von-Guericke-University Magdeburg, Germany
{miotto,friesen,roesner}@ovgu.de

Abstract. Wizard of Oz (WOZ) systems and WOZ experiments are an important tool for basic and applied research in HCI. We report about using SEMAINE as a flexible component based middleware with a loose coupling of components as software infrastructure for WOZ experiments in human companion interaction. We focus on our experimental WOZ designs, their realisation within the SEMAINE framework and lessons learned from deploying the implemented solutions as the basis for ongoing controlled experiments with 120 subjects.

Keywords: Wizard of Oz, Companion Systems, Emotion, Multimodal, Message Oriented Middleware, SEMAINE.

1 Introduction

A Companion System (CS) can be described as *"an agent or 'presence' that stays with the user for long periods of time, developing a relationship and 'knowing' its owners preferences and wishes. The Companion communicates with the user primarily through speech, but also using other technologies such as touch screens and sensors."* [3]. Companion systems will assist their users in managing their daily life. An elderly person that may still live on his own with such a technical support is more likely envisaged as a companion user than a young professional. So support in mundane activities rather than in specialised business applications is of high relevance.

In order to explore companion systems, Wizard of Oz (WOZ) experiments are a usual approach [5,6,11]. The WOZ concept has a common usage in the fields of experimental psychology, human factors, ergonomics, linguistics, and usability engineering to describe a testing design methodology where an experimenter ("Wizard") simulates the behaviour of a theoretical intelligent computer application.

Different software components communicate with each other, thus the system needs an infrastructure that supports flexibility and reusability. Message Oriented Middleware (MOM) is a software infrastructure with the focus on sending and receiving messages between different systems [2]. In general MOM supplies a base for application interoperability in heterogenous and complex environments. Application Programming Interfaces (APIs) which are distributed across diverse

J.A. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2011, LNCS 6761, pp. 121–130, 2011.

platforms and networks are typically provided by a MOM. This facilitates easy integration of components running on different operating systems and written in different programming languages. The SEMAINE API provides an abstraction layer over a MOM and provides the needed functionality for modular systems dealing with emotions [9].

In this paper we suggest a framework, with which WOZ experiments can be provided in flexible way.

This paper is organized as follows. We first introduce the Wizard of Oz paradigm in section 2. Section 3 describes our experimental design. In section 4 we present the architecture of the framework followed by an description of the implemented interface components in Section 5. Section 6 concludes the paper and provides some insights on our future work.

2 Wizard of Oz

Wizard of Oz (WOZ) systems are an important tool for basic and applied research in Human-Computer Interaction (HCI) and thus likewise for companion systems. This is especially true when WOZ experiments are interleaved with the development and implementation of additional system functionality, e. g. development and/or integration of autonomous and automatic components that replace functionality formerly provided by the wizard.

A WOZ system allows the observation of a user operating an apparently fully functioning system whose missing services are supplemented by a hidden wizard. The user is not aware of the presence of the wizard and is led to believe that the computer system is fully operational. The wizard observes the user through a dedicated computer system connected to the observed system over a network. When the user invokes a task that is not available in this observed system, the wizard simulates the effect of the task. Through the observation of users behavior, designers can identify users needs when accomplishing a particular set of relevant functions and evaluate the particular interface used to accomplish the functions. During a WOZ experiment, the exchanged data between the observed user and the wizard is recorded for further analysis [8].

3 Experimental WOZ Design

By definition a companion is with his user for a long period of time. So certain aspects of the human companion interaction will only be adequately dealt with in long term investigations with developed companion systems that users can carry with them and make use of in their daily life. For our experimental design we had to take into account that such developed systems do not yet exist. Even more some aspects of the interaction with a companion have to be investigated before such systems can be designed and implemented. In our experiment we investigate intentions ascribed to the companion system by users. Further description of this paradigm can be found in [7].

3.1 The Cover Story

Our cover story that is told to the subjects in the wizard of oz experiments takes these constraints into account: The subjects are told that they will run through the personalisation phase of a companion system – a phase that is a prerequisite for the system to be later adapted to the individual user characteristics and preferences – and that they therefore both have to answer a number of questions as well as to succesfully complete some tasks. Another aspect of this cover story is the resulting personal involvement of the subjects that is additionally supported by addressing the subject with his name and displaying the name as running head in various screens.

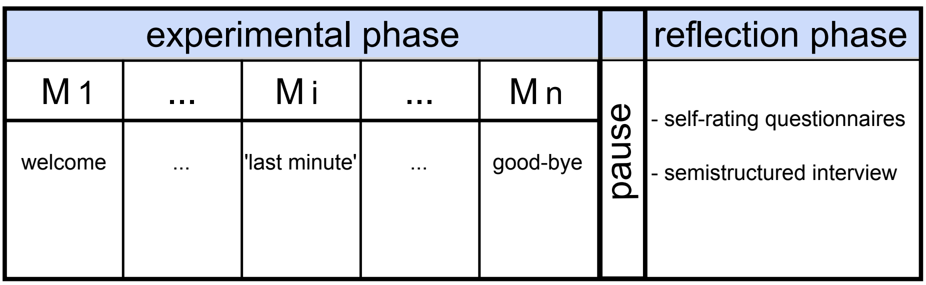

Fig. 1. Design of the WOZ experiment

The overall design of our WOZ experiment comprises an experimental and a reflection phase. The experimental phase was divided into different modules (c. f. Fig. 1). Each module stands for a part in the experimental phase and can employ different software components.

The module named 'last minute' was designed to investigate how users interact with a companion system in a mundane situation with the need for planning, replanning and strategy change. The subject interacts only vocally with the system.

3.2 Last Minute

The user is asked to imagine a situation in summer with most of his friends already in holidays when s/he as a surprise gets informed to have won a two weeks vacation. The prize includes as well the opportunity to select all necessary items for the travel from an online catalogue organized in a number of categories (e. g. coats, trousers and skirts, hats, underwear, sports equipment, etc.). For this selection fifteen minutes are available, because then the taxi to the airport will arrive and the packed suitcase will be available at check in.

4 Architecture of the Framework

The aim of our research was the development of an instrument for the realization of WOZ experiments, which fulfills the following requirements:

1. simple, easy and efficient development as well as integration of modules, software components
2. and distributed WOZ experiments across different locations.

The software for the WOZ experiments is component based and thus modular structured. The components communicate using a MOM which is specifically designed to integrate different applications or processes through messages being routed from publishers to subscribers. The aim of this "glue technology" is to glue together applications both within and across organizations, without having to reengineer individual components. An important advantage of a generic Message Oriented Middleware lies in its flexibility. The system architecture can be rearranged very easily – adding or removing a component consuming or creating a specific message type does not require any changes elsewhere in the architecture [2].

In the first implementation OAA (Open Agent Architecture) [4] was used as MOM but this platform was replaced by the SEMAINE API [9]. The SEMAINE API provides an abstraction layer over a MOM that allows the components to deal with messages in a type-specific way. The MOM currently used in the SEMAINE API is ActiveMQ[1] from the Apache project.

Our approach uses the SEMAINE API, which is an open source framework for building emotion-oriented systems and provides Java and C++ wrappers around a MOM. This framework allows the communication between components running on different operating systems via XML and other standardized formats. Different components can run on physically distinct hardware as long as they are connected within a network.

5 Implementation

The interface components of the module 'last minute' are shown in Fig. 2. This 'last minute' module uses a control interface for the wizard ("wizard interface"), an visual user interface for the subject and a text-to-speech system for the speech synthesis ("subject interface"). The experiment is controlled by the Wizard via the control interface. From here the Wizard can select text blocks for the speech synthesis and control the visual operational sequence on the monitor of the test person [5].

An example interaction:
1. The subject says he would like to pack items (e. g. "I want to pack two t-shirts").
2. The Wizard understands the utterance, selects the equivalent term on the monitor and thus confirms the selection.

[1] http://activemq.apache.org/

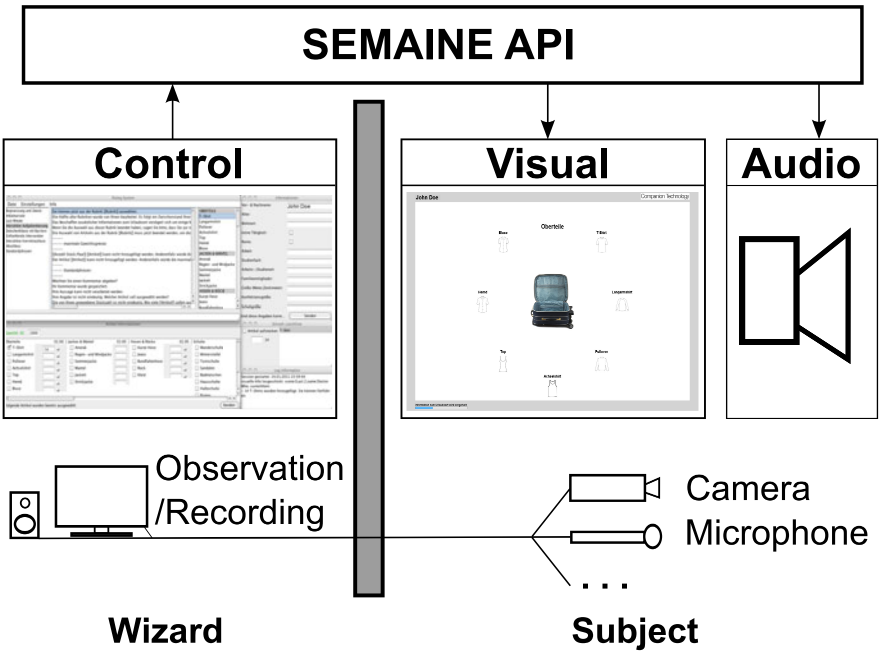

Fig. 2. Interfaces of the 'last minute' module

3. The SEMAINE API sends this information to the visual user interface and to the speech synthesis component.
4. The visual user interface displays the new information and the speech synthesis component generates the respective language output (e. g. "two t-shirts were added").

The WOZ system used is based on three major components integrated using the SEMAINE API. These are the control interface component of the Wizard and the visual user interface and speech output component of the test person. In the following we will describe these implemented software components in more detail.

5.1 Wizard Control Interface

The wizard control component and its interface is shown in Fig. 3. It provides typical features like a configuration menu and a log window. The frames for the different tasks are arranged as follows.

The dialog control frame contains the selection of modules of the WOZ experiment (1), predefined text patterns (2), the item selection (3) and at the bottom a text field for free text input. For each module of the WOZ experiment selected all predefined text patterns are loaded and displayed. Some text patterns need additional information (e. g. which item should be packed or which catalogue section to load). After selecting a text pattern and – optional – an item the sentence

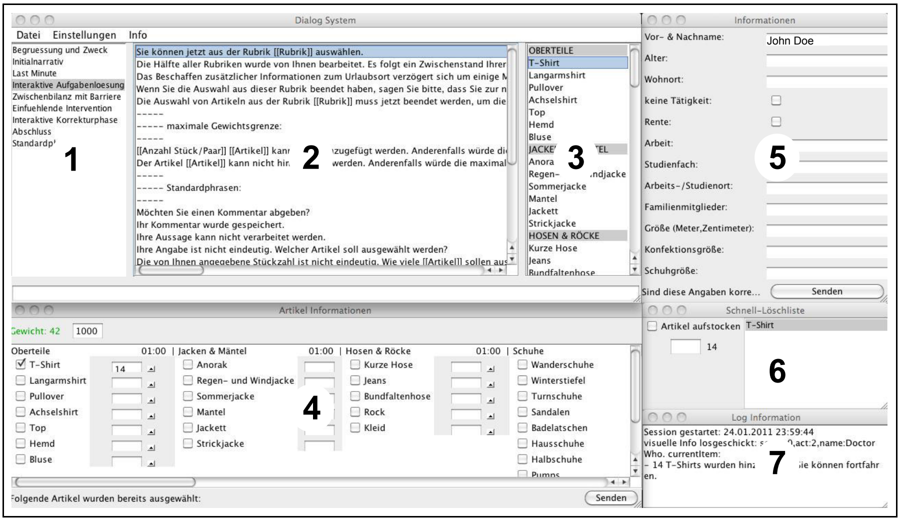

Fig. 3. Wizard Control Component

can be send to the other components by pressing 'enter' or with a double-click. If a sentence is needed but can not be found fast enough or is not available in the text pattern list, it can be typed into the free text field (providing auto completion) and sent. The Article information frame (4) provides an overview of all items already in the suitcase and their actual weight. It can also be used to pack items. The Information frame (5) provides the subjects personal information and is used in the personalization stage of the experiment. The Fast-Erase frame (6) shows all packed items in alphabetical order and is used to unpack items or change their amount quickly. The Log frame (7) shows a log of all events and sent messages to support the observation of the subjects screen. The Frames 1 - 6 support fast handling and provide easy sending functionality and hotkeys.

5.2 Visual User Interface

The visual user interface is started seperately but is controlled by messages sent from the wizard control component. It visualizes information about the experiment and the selection menus during interaction (c. f. Fig 4) as well as the subjects name on top and additional information about the session state. With this component the subject is able to observe his actions on the screen. The visual user interface is written in JavaFX[2].

5.3 Speech Component

The Speech Component receives messages send from the wizard control component and sends the messages to a MARY TTS[3] server to synthesise these

[2] http://javafx.com/

[3] http://mary.dfki.de/

Fig. 4. Visual User Interface: a selection menu

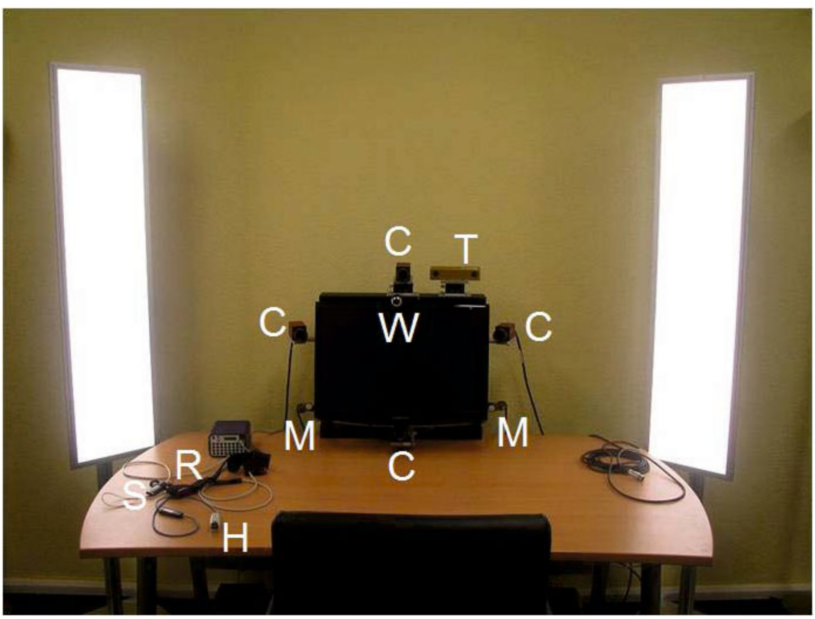

Fig. 5. The experimental hardware setting. C=High resolution camera, H=Heart beat clip, M=Microphone, R=Respiration belt, S=Skin conductance clip, T=Stereo camera, W=Observation webcam. Not in the picture: Headwear microphone.

messages. The MARY TTS platform is an open source and modular architecture for building text-to-speech systems [10]. It supports various pronounciation features of synthesised speech and will be employed to simulate emotionally sounding voice (e. g. friendly, sad, angry, . . .) in the ongoing improvement of the system.

5.4 Experimental Setting

The environment for the subjects (c. f. Fig. 5) is as follows: The subjects sit comfortable on a chair at a desk. They are told to lean back and are instructed not to move the left hand because the skin conductance and heart beat are measured at the left fingers. For respiration recording the subjects get an elastic belt around the chest. Sound is recorded with two microphones next to the screen and one headwear microphone. Video is recorded with one observation webcam, four high resolution cameras and one stereo camera. Illumination comes from two light panels next to the desk and three light panels at the ceiling. The window of the room is shut, so no daylight (constant illumination) and no noise comes in.

6 Conclusions and Future Work

In this paper we have proposed a software design for Wizard of Oz experiments that uses the SEMAINE API as the platform for the components' communication. This design allows flexibility for adding new components and exchange and reuse of components. The implemented software is used to perform experiments with 120 subjects and will be used – with different component constellations – for further experiments.

Automating parts of the WOZ experiment is work in progress. The architecture for automating the spoken dialogue between system and user is designed as follows. The software is divided into four SEMAINE components.

- One component connects a commercial speech recognizer (ASR) and text to speech (TTS) software[4] to the SEMAINE system. This component will from now on be referred to as the connector component.
- The container component acts as a storage for items which the user has packed into his luggage.
- A third component is called the control component and its task is to send commands concerning the insertion or removal of luggage items into the container, as well as keeping track of the active category for selection and notifying other components of which items are currently available for packing/removal.
- The last component is the grammar component based on Grammatical Framwork (GF) [1], which has two tasks. The first task is parsing utterances based on GF recognized by the ASR and transforming them into commands related

[4] http://www.nuance.com/

to packing/removing items from the luggage. These commands are then sent to the control component for further processing and, if applicable, to the container component, too. The second task of the grammar component is the reception and processing of status updates of the container. These status updates reflect successful operations inside the container. After such a status update the grammar component transforms the confirmation message into a natural language sentence, and sends this linearization to the text to speech component.

In addition to automating functionality of the wizard, future work includes the development of a formal description for the software components and experimental modules, e. g. by XML. Using this description several experimental settings (e. g. with more and different experimental modules, interfaces and components) can be created easily and with high flexibility.

Acknowledgements

The work reported is performed within the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG)[5]. The responsibility for the content of this paper lies with the authors.

References

1. Angelov, K., Bringert, B., Ranta, A.: PGF: A Portable Run-time Format for Type-theoretical Grammars. J. of Logic, Lang. and Inf. 19, 201–228 (2010), `http://dx.doi.org/10.1007/s10849-009-9112-y`
2. Banavar, G., Chandra, T., Strom, R., Sturman, D.: A Case for Message Oriented Middleware. In: Jayanti, P. (ed.) DISC 1999. LNCS, vol. 1693, pp. 1–17. Springer, Heidelberg (1999)
3. Catizone, R., Worgan, S.F., Wilks, Y., Dingli, A., Cheng, W.: A world-hybrid approach to a conversational Companion for reminiscing about images. In: Close Engagements with Artificial Companions: Key Social, Psychological, Ethical and Design Issues, pp. 157–168. John Benjamins, Amsterdam (2010)
4. Cheyer, A., Martin, D.: The Open Agent Architecture. Autonomous Agents and Multi-Agent Systems 4(1-2), 143–148 (2001), `http://dx.doi.org/10.1023/A:1010091302035`
5. Kelley, J.F.: An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications. ACM Trans. Inf. Syst. 2(1), 26–41 (1984)
6. Legát, M., Grůber, M., Ircing, P.: Wizard of Oz Data Collection for the Czech Senior Companion Dialogue System. In: Fourth International Workshop on Human-Computer Conversation, pp. 1–4. University of Sheffield, Sheffield (2008)
7. Rösner, D., Friesen, R., Otto, M., Lange, J., Haase, M., Frommer, J.: Intentionality in Interacting with Companion Systems – an Empirical Approach. In: Jacko, J.A. (ed.) HCII 2011. LNCS, vol. 6761. Springer, Heidelberg (2011)

[5] `http://www.sfb-trr-62.de/`

8. Salber, D., Coutaz, J.: Applying the Wizard of Oz Technique to the Study of Multimodal Systems. In: Bass, L.J., Unger, C., Gornostaev, J. (eds.) EWHCI 1993. LNCS, vol. 753, pp. 219–230. Springer, Heidelberg (1993)
9. Schröder, M.: The SEMAINE API: Towards a Standards-Based Framework for Building Emotion-Oriented Systems. In: Advances in Human-Computer Interaction, vol. 21, pp. 319–406 (2010)
10. Schröder, M., Pammi, S.C., Türk, O.: Multilingual MARY TTS participation in the Blizzard Challenge 2009. In: Proceedings of Blizzard Challenge 2009. o.A. (2009)
11. Webb, N., Benyon, D., Bradley, J., Hansen, P., Mival, O.: Wizard of Oz Experiments for a Companion Dialogue System: Eliciting Companionable Conversation. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association, ELRA (2010)