

WOOF: EIN FRAMEWORK FÜR WIZARD OF OZ EXPERIMENTE

Markus Huber¹ Werner Meyer² Kati Nowack² Peter Geßler²

¹ InnoTec21 GmbH ² BTU Cottbus-Senftenberg
markus.huber@innotec21.de

Zusammenfassung: Im Rahmen des Industrieprojekts *Universal Cognitive User Interface* (UCUI) entsteht ein Framework zur Erstellung und Durchführung von Wizard of Oz Experimenten. Dadurch kann eine nutzergetriebene Systemkonstruktion unterstützt werden, bei der aufeinander aufbauende Systemsimulatoren, die nach und nach erweitert werden können, benötigt werden.

Der Text beschreibt die Anforderungen an das Framework, dessen Aufbau und seine grundlegende Funktionsweise. Ein Beispiel eines Simulators ist ebenso enthalten.

1 Einleitung

„Man kann nicht nicht kommunizieren [...]“, schrieb Paul Watzlawick [5] Mitte des letzten Jahrhunderts, „[...] denn jede Kommunikation [...] ist Verhalten und genauso wie man sich nicht nicht verhalten kann, kann man nicht nicht kommunizieren.“

Trotzdem fallen die meisten heutigen Benutzerschnittstellen durch eintöniges und unsinniges Verhalten eher unangenehm auf. Was sie damit – auch über ihre Macher – kommunizieren, steht auf einem anderen Blatt. Allerdings sind gerade natürlichsprachliche Benutzerschnittstellen schwer umzusetzen, da Sprache schier endlose Möglichkeiten bietet, um sich auszudrücken.

Damit eine Benutzerschnittstelle angemessen reagieren kann, muss sie einerseits die syntaktischen Unterschiede der Eingaben in einer höheren Semantik auflösen und andererseits wissen, wie sie – auch im Dialog mit dem Benutzer – über diese Semantik zum gemeinsamen Ziel kommen kann. In diesem Sinne muss sie also über Kognition verfügen.

Ein oft beschrittener Weg besteht nun darin, Probanden vor eine Simulation einer solchen Benutzerschnittstelle zu setzen und die Interaktionen aufzuzeichnen. Danach versucht man, die entstandenen Dialoge nachzubauen, damit das System so Angemessenheit simulieren kann.

Im Umfeld von grafischen Benutzerschnittstellen existieren sog. Mock Up Frameworks, mit denen das Aussehen und die Abfolge von Fenstern und Dialogen simuliert werden kann. Im Bereich kognitiver Benutzerschnittstellen gibt es unseres Wissens nach keine derartigen Frameworks, die vor allem auch die im Rahmen des UCUI Projekts gestellten Anforderungen erfüllt hätten.¹ Aus diesem Grund entstand das hier vorgestellte Wizard of Oz Framework.

Das UCUI Projekt wird im folgenden Abschnitt kurz vorgestellt. Im Abschnitt 3 wird noch einmal genauer auf die Methode der Simulation eingegangen, um schließlich das Framework im Abschnitt 4 zu besprechen. Der Text schließt mit einer Zusammenfassung und einem Ausblick.

2 Universal Cognitive User Interface

Im Projekt *Universal Cognitive User Interface* (UCUI) soll ein System realisiert werden, das in der Lage ist durch die Eingabe eines Nutzers eine entsprechende Handlung über die an diesem System angeschlossenen Geräte im Haushalt zu tätigen. Zur Steuerung des Systems kann der

¹Einzig WebWOZ von der School of Computer Science & Statistics des Dubliner Trinity College kam unseren Vorstellungen sehr nahe.

Anwender eine Eingabe über Sprache, Gestik oder Tastatur vornehmen, wodurch eine barrierefreie Anwendung ermöglicht werden soll. Dabei soll das System eigenständig handeln, ohne umfangreiche Datensammlungen (Big Data) einzusetzen, die zudem auch eine explizit ausgeschlossene Internetverbindung benötigen würden. Durch den Verzicht auf die Datensammlungen und den Kontakt nach außen soll auch der Datenschutz und damit die Privatsphäre des Nutzers gewährleistet werden. Die vom System gewonnenen nutzerspezifischen Daten werden mit Hilfe einer kognitiven Verhaltenssteuerung verarbeitet, um sich an das Kommunikationsverhalten des Nutzers sowie seine Problemlösungsstrategien anzupassen. Ziel ist es, dass das System sich auf den Nutzer einstellt und nicht umgekehrt; auch unter der Berücksichtigung, dass es von Personen eingesetzt wird, die weniger im Umgang mit technischen Geräten geübt sind. Zugleich werden die nutzerspezifischen Daten nicht an andere Nutzer weitergegeben, um eventuelle Rückschlüsse auf das Verhalten der Anwender zu vermeiden (e.g., [6, 4]).

Um ein angemessenes Verhalten des UCUI Systems zu gewährleisten, ist es wichtig, mögliche Mensch-Maschine-Interaktionen in das System einzubinden. Da bspw. bei der Sprachsteuerung unterschiedliche Eingaben dieselbe Bedeutung besitzen können, muss ermittelt werden, durch welche Äußerungen das System auf Grundlage einer semantischen Verarbeitung eine angemessene Reaktion ausübt. Die semantische Verarbeitung führt alle Ein- und Ausgabemodalitäten auf einer Ebene zusammen.

Zur Ermittlung der vielfältigen Eingabemöglichkeiten eines Nutzers, ist die Durchführung von Wizard of Oz Experimenten geeignet. Das Anwenderverhalten gegenüber dem System kann damit genauer untersucht und die gesammelten Ergebnisse in das zukünftige System integriert werden.

3 Wizard of Oz Experimente

Die Konstruktion des UCUI wird nicht technik-, sondern nutzergetrieben erfolgen. Dies stellt in Anbetracht eines noch nicht vollständig entwickelten Systems eine besondere Herausforderung dar. So sollen Funktionen bereits vor deren Implementierung auf ihre Bedienbarkeit hin evaluiert und, wenn nötig, optimiert werden. Zu diesem Zweck kommt im Projekt die so genannte *Wizard of Oz* Methode zum Einsatz.

Hauptmerkmal eines Wizard of Oz Experiments ist die Simulation eines fertigen Systems durch einen Menschen (den Wizard). Benutzer interagieren während des Experiments mit dem Interface eines vermeintlichen Systems. Die Systemreaktionen werden jedoch in Wirklichkeit durch den Wizard simuliert und durch das Interface an den Benutzer zurückgemeldet. Damit Testpersonen nicht den Verdacht schöpfen, dass es sich nicht um ein computergesteuertes System handelt, ergeben sich für die Wizard of Oz Methode eine Reihe von Anforderungen (e.g., [1, 2]).

Laut [1] muss für ein erfolgreiches Wizard of Oz Experiment erstens das zukünftige System in Anbetracht menschlicher Einschränkungen simulierbar sein, zweitens das Verhalten des zukünftigen Systems spezifiziert werden und drittens das System glaubwürdig simuliert werden.

So muss der Wizard in sehr kurzer Zeit angemessen und fehlerfrei auf die Nutzereingaben reagieren. Dies kann bspw. durch verschiedene vorgefertigte, häufig benutzte, Rückmeldungen mit Schnellzugriff (z.B. *Bitte warten, Ihre Anfrage wird bearbeitet.*) erreicht werden sowie durch Training des Wizard, um fehlerfreie und konstante Reaktionen zu gewährleisten.

Ein weiterer Punkt ist der Einsatz von klaren Szenarien und Aufgaben, bei denen das Bedienzziel bekannt ist und auch nur mithilfe des Systems erreicht werden kann, ohne jedoch die Benutzer in ihren Lösungswegen, verbalen Äußerungen und Gesten einzuschränken (e.g., [1, 2]). Der Prozess der Aufgabenkonstruktion sollte somit berücksichtigen, welche Interaktionsmöglichkeiten Benutzer mit dem System haben sollten und diese auch klar den Probanden mitteilen (vgl. [2]). Im Rahmen des hier vorgestellten UCUI Projektes wird dies mittels schriftlicher

Handlungsinstruktionen und der Vorführung des Systems anhand einer Beispielanwendung (ein einfacher Getränkeautomat) durch den Versuchsleiter – nicht den Wizard – realisiert. Ferner sollte die Aufgabenerstellung auf Hypothesen bezüglich des Benutzer- (i.e., Wie werden Benutzer reagieren?) und Systemverhaltens (i.e., Wie soll das System reagieren?) aufbauen (vgl. [2]).

Zuletzt sollte eine erfolgreiche, nutzergetriebene Konstruktion auf mehreren Wizard of Oz Experimenten aufbauen, bei denen das zu entwickelnde System zunehmend autonom mit dem Benutzer interagiert und weniger durch den Wizard simuliert wird (e.g., [3]). Dies wird im Rahmen des UCUI Projektes mittels drei aufeinander aufbauender Versuchsreihen mit verschiedenen Systemsimulatoren erreicht, an deren Ende die Evaluation des optimierten Gesamtsystems steht.

4 WoOF: Wizard of Oz Framework

Um die nutzergetriebene Konstruktion mit verschiedenen Simulatoren unterstützen zu können, wurde mit dem *Wizard of Oz Framework* (WoOF), die Möglichkeit geschaffen, unterschiedliche, erweiterbare Systemsimulatoren zu erstellen und auszuführen. Im folgenden Unterabschnitt werden die Anforderungen aufgeführt, die an das WoOF gestellt werden, bevor im Unterabschnitt 4.2 seine Umsetzung vorgestellt wird. Darin wird auch der Systemsimulator, wie er in der ersten Experimentierphase des Projektes benutzt wird, beschrieben.

4.1 Die Anforderungen

Die hier zusammengetragenen Anforderungen stellen teilweise allgemeine Anforderungen, denen ein Systemsimulator-Framework für Wizard of Oz Experimente genügen sollte, dar. Es fließen auch allgemeine und spezielle Überlegungen aus den Abschnitten 2 und 3 mit ein.

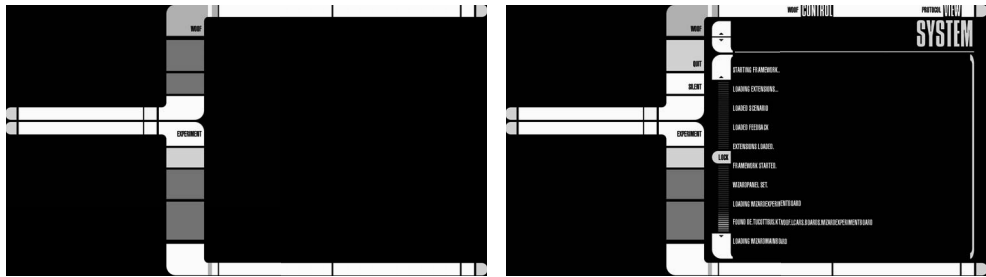
Da ein echtes System zu simulieren ist, das visuelle und auditive Rückmeldungen an den Benutzer geben soll, muss zum Einen die Möglichkeit vorhanden sein, visuelle Objekte auf einem Monitor darzustellen und zum Anderen muss es möglich sein, Audiodaten an den Benutzer zu übertragen. Das fertige System soll sowohl über Touch- als auch über Spracheingabe gesteuert werden können, wodurch der Benutzer mit den visuellen Objekten interagieren können muss und Audiodaten auch vom Benutzer zum Simulator übertragen werden müssen.

Neben diesen grundlegenden Anforderungen, muss eine gute Unterstützung der Durchführung der Experimente gewährleistet sein. Dies umfasst einerseits die Erstellung von Systemsimulatoren sowie die Funktionen für den Wizard während der Experimente, andererseits die Aufzeichnung der einzelnen Experimente und die Speicherung der anfallenden Daten.

Wie in Abschnitt 3 bereits angesprochen, sollen einzelne Szenarien festgelegt werden können, die als Aufgaben an die Probanden zu verstehen sind. Zwischen diesen Aufgaben muss gewechselt werden können. Ein Szenario ist dabei als eine Einheit aus Aufgabenstellung, Bedienzweck und möglichen visuellen und auditiven Rückmeldungen zu verstehen.

Als Ergebnis der Experimente soll eine Sammlung von Anwenderverhalten gegenüber dem System entstehen. Daraus ergibt sich, dass die Interaktionen mit dem System aufgezeichnet werden müssen. Dazu sind der Bildschirminhalt, den die Probanden sehen, wie auch die Touch-Ereignisse, die sie auf dem Monitor auslösen und darüber hinaus die Spracheingaben, die sie tätigen, zu speichern. Da später im UCUI Projekt auch Gesten als Eingabemedium Verwendung finden sollen, muss auch ein Videomitschnitt der Probanden erstellt werden. Um die Daten später leichter auswerten zu können, sollte auch die Sprachausgabe des Systems gespeichert werden.

Damit die Daten pro Proband gespeichert werden können, muss weiterhin eine Sitzungsverwaltung vorhanden sein. Die Gewährleistung des Datenschutzes der Teilnehmer der Experimente stellt ausdrücklich keine Anforderung an das Framework dar. Diese Aufgabe muss von den Versuchsleitern übernommen werden.



(a) Startbildschirm.

(b) Systemsteuerung mit Startprotokoll.

Abbildung 1: WoOF ohne geladenes Experiment.

4.2 Die Umsetzung

Die Implementierung des WoOF orientiert sich an seinem primären Einsatzort, dem *Cognitive Systems Lab* (CSL) des Lehrstuhl für Kommunikationstechnik an der BTU Cottbus-Senftenberg. Für die dortige Audio-Hardware sind in Java geschriebene Treiber vorhanden. Alle grafischen Oberflächen im Labor setzen auf LCARSWT auf, das ebenfalls in Java erstellt ist. Deswegen lag die Benutzung von Java als Programmiersprache für das Framework nahe.

4.2.1 Die Benutzeroberfläche

Bei LCARSWT [7] handelt es sich um ein Widget Toolkit, mit dem Oberflächen erstellt werden können, die sich optisch an das u.a. aus der Fernsehserie *Star Trek: The Next Generation* bekannte LCARS² anlehnen. Das wesentliche Merkmal derartiger Oberflächen neben der reinen Touch-Bedienung³, ist das völlige Fehlen von Fenstern und Dialogen. Sie sind vielmehr als digitales Pendant zu Schaltbrettern zu sehen.

Daraus ergibt sich, dass eine Oberfläche als ein aus funktionalen Einheiten bestehendes Ganzes betrachtet wird, das in LCARSWT als *Panel* bezeichnet wird. Die Einheiten müssen nicht notwendigerweise zusammenhängende Bereiche sein, noch müssen sie eine bestimmte Form haben. Diese Eigenheit wird im WoOF allerdings nur sehr behutsam eingesetzt.

In Abbildung 1 ist neben dem Startbildschirm des Framework die zentrale Steuereinheit zu sehen. In ihr kann man die verschiedenen Protokolle nachlesen, das System verlassen und das Audio-Feedback der Oberfläche aktivieren bzw. deaktivieren. Man erkennt bereits eine funktionale Einheit. Grob gesprochen, entsprechen drei dieser Einheiten den schwarzen Bereichen.

Die Oberfläche bietet insgesamt vier funktionale Einheiten, die zusammen die Steuerung eines Systemsimulators ermöglichen. Drei von diesen Einheiten werden mit sog. *Boards* gefüllt. Dem großen rechten Bereich aus Abbildung 2 stehen neben der frei füllbaren Fläche die beiden Schaltflächen unter der Schaltfläche *WOOF* und die oberen beiden Titelbereiche zur Verfügung. Für die kleinen linken Bereiche stehen neben den Flächen je eine Statuszeile nebst einem Quadrat, das blinken kann, zwischen den beiden Bereichen zur Verfügung. Die vierte Einheit kann mit einem sog. *Controller* belegt werden. Für diese Einheit steht keine füllbare Fläche zur Verfügung, sondern lediglich die drei Schaltflächen unter der Schaltfläche *EXPERIMENT*, das darunter liegende Statusfeld, das auch blinken kann sowie die unteren beiden Titelbereiche. Für alle Schaltflächen gilt, dass sie sowohl als normale, als auch als einrastende oder als blinkende Schaltflächen betrieben werden können.

²Die Abkürzung steht für *Library Computer Access and Retrieval System* (s. <http://www.lcars.org.uk/>).

³Wobei der geneigte Trekkie einwenden wird, dass LCARS Oberflächen auch und vor allem in Kombination mit Sprache benutzt werden. Da allerdings noch kein UCUI existiert, verzichtet WoOF auf eine Sprachsteuerung.

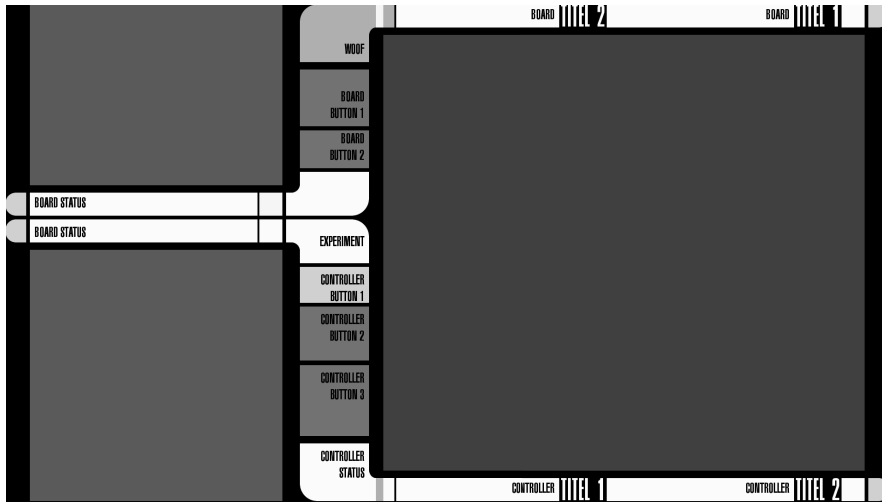


Abbildung 2: Die großen farbigen Flächen können mit *Boards* belegt werden.

Die vier funktionalen Einheiten können jeweils mit beliebig vielen Instanzen belegt werden, wobei immer nur eine sichtbar sein und damit die Kontrolle über die immer sichtbaren Elemente des *Panel*, die sich alle Instanzen teilen, ausüben kann. Intern existiert für jede Einheit ein Stack, sodass eine Instanz auftauchen und wieder verschwinden kann, um danach die gleiche Oberfläche zu hinterlassen.

Jedes *Board* wiederum besteht aus *Controls* aus einer erweiterbaren Sammlung und kann auch beliebige weitere LCARS Elemente aufnehmen. Um ein möglichst einheitliches Erscheinungsbild zu gewährleisten, wurde eine kleine eigenständige Bibliothek⁴, die auf LCARSWT aufsetzt und einige Widgets wie Rahmen, Textfelder, Listen, Eingabefelder und Videoviewer anbietet, erstellt. Sie ermöglicht es auch, den Bildschirm in einzelne – nicht notwendigerweise disjunkte – Boxen zu unterteilen, um *Controls* und Elemente einfacher positionieren zu können. Die grundlegende Funktionalität, um *Boards* anzeigen und verstecken zu können, ist ebenfalls hier angesiedelt. Eine weitere Teilbibliothek, um Bereiche zu dekorieren, ist in Anfängen enthalten.

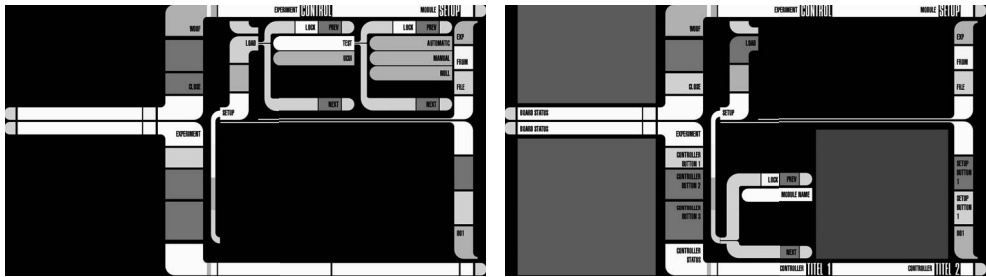
Die gesamte Oberfläche ist auf die Benutzung mit einem Touch-Bildschirm ausgelegt. Es ist aber auch möglich, Eingaben per Tastatur zu machen. Einige Möglichkeiten im Zusammenhang mit Touch-Bedienung, die LCARSWT durch eine kürzlich erfolgte Neuerung bietet, werden allerdings noch nicht unterstützt.

4.2.2 Das Framework

Das Framework unter der Oberfläche besteht zunächst aus der zentralen Klasse *WoOF*, die als Singleton implementiert ist. Das einzig erzeugbare Objekt dieser Klasse verwaltet verschiedene Ressourcen und stellt den Zugriff auf unterschiedliche Dienste bereit. Nach dem Start des Systems, lädt *WoOF* grundlegende Einstellungen aus einer Konfigurationsdatei, legt das erste Protokoll an und initialisiert eventuell vorhandene Erweiterungen. Dabei handelt es sich um Klassen, die keine einheitliche Schnittstelle bereit stellen und direkt im Code von Systemsimulatoren benutzt werden. Sie gehören nicht zur Kernfunktionalität des Framework, sollen aber allen Simulatoren zur Verfügung stehen. Danach kann ein Experiment geladen werden.

Experimente werden in Projekten zusammengefasst. Auf diese Weise kann man gemeinsam

⁴Sie heißt *pmllh*, was für *poor man's LCARS layout helpers* steht.



(a) Laden eines Experiments.

(b) Bereich für Einstellungen.

Abbildung 3: Auswahl und Einstellen eines Experiments.

benutzte Einstellungen und Ressourcen zur Verfügung stellen. Alle Einstellungen werden aus Konfigurationsdateien geladen, die dem gleichen Format genügen. Einem Schlüssel wird durch ein Gleichheitszeichen ein Wert zugewiesen, der bis zum Ende der Zeile bzw. dem Kommentarzeichen # reicht. WoOF stellt für die Abfrage der Werte einige Methoden zur Verfügung, sodass neben einfachen Zeichenketten auch Wahrheitswerte und Listen möglich sind.⁵ Systemweite Einstellungen werden von projektweit gültigen und diese wiederum von den Einstellungen für ein Experiment überschrieben; Listen können auch an beiden Enden erweitert werden.

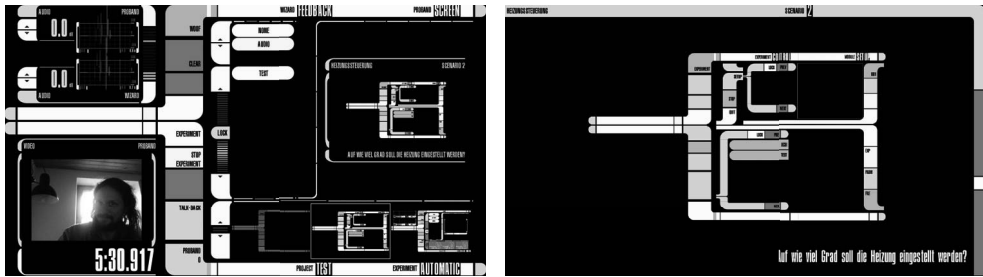
In Abbildung 3a sieht man das *Board* zum Laden eines Experiments nach Betätigen der Schaltfläche *EXPERIMENT*. Über dieses *Board* kann man ein Experiment auch wieder schließen und ein anderes laden. Neben den *Boards* und *Controllers* besteht ein Experiment aus einer Beschreibung und einer Liste von Modulen, die zu laden sind. Je nach benutzten Bestandteilen stehen weitere Einstellungsmöglichkeiten in den Konfigurationsdateien zur Verfügung.

Module sind Klassen von denen jeweils ein Objekt beim Laden eines Experiments erzeugt und registriert wird. Sie stellen bestimmte Funktionen bereit, werden aber nicht direkt im Code von Systemsimulatoren benutzt. Sie können zusammen mit den weiteren Mechanismen eingesetzt werden, die der Erweiterbarkeit des Framework dienen, haben aber auch ohne sie ihre Existenzberechtigung. Module können auch Einstellungen über sog. *Setups* zur Verfügung stellen, die dann zur Laufzeit angepasst werden können. In Abbildung 3b sieht man rechts unten einen Bereich, der für diese Einstellmöglichkeiten zur Verfügung steht. Die beiden mittleren Schaltflächen rechts daneben gehören ebenfalls zu dieser funktionalen Einheit. Über die Liste links davon kann man die einzelnen Module auswählen.

Die angesprochenen weiteren Mechanismen sind das Konzept der sog. *Function-Provider* und *Pipelines* zur Datenverarbeitung. Bei letzteren handelt es sich um eine einfache Implementierung des *Pipes and Filters*-Musters⁶, bei dem Daten durch eine Pipeline geschickt werden, wobei angemeldete Filter sie auf ihrem Weg verändern können. Objekte melden sich bei WoOF als Erzeuger, Filter oder Verwerter für Daten eines speziellen Typs an. Die *Function-Provider* ermöglichen es, allgemeine Funktionen, die als Schnittstellen vorliegen, zu benutzen, ohne das konkrete Objekt, das diese Funktionen zur Verfügung stellt, kennen zu müssen. Objekte können WoOF mitteilen welche Funktionen sie bereitstellen möchten. Der Aufruf passiert über statische Methoden der Schnittstellen und wird von WoOF abgewickelt. So ist es möglich bspw. die Ausgabe von Audio-Daten durch verschiedene Module vornehmen zu lassen, sodass man das Framework an unterschiedliche Hardware anpassen kann, je nachdem welches Modul geladen wurde. Soll eine Funktion von maximal einem Objekt bereitgestellt werden, können sog. *exclusive Function-Provider* benutzt werden, die eine Fehlermeldung auslösen, wenn ein zweites

⁵Weitere Datentypen sind für spätere Versionen geplant.

⁶Diese eigenständige Bibliothek nennt sich *pmpaf*, was für *poor man's Pipes and Filters* steht.



(a) Ansicht für den Wizard.

(b) Ansicht für den Proband.

Abbildung 4: Das erste Experiment im UCUI.

Objekt die Funktion übernehmen möchte.

WoOF bietet eine einfache Sitzungsverwaltung an, die sicherstellt, dass ein Experiment mit einem neuen Probanden erst gestartet werden kann, wenn die Sitzung konfiguriert ist. Dazu müssen drei Parameter festgelegt werden, aus denen sich dann auch der Speicherort der Daten ergibt. Diese Parameter können über ein spezielles *Board* eingegeben oder automatisch über ein Modul gesetzt werden. Für die Sitzungsdaten existiert eine eigene Pipeline, bei der nur WoOF als Erzeuger und Verwerter auftreten kann, aber beliebige Objekte als Filter agieren können. Somit können auch halbautomatische Varianten mit einem Modul und einem *Board* umgesetzt werden.

Das Framework läuft größtenteils ereignisgesteuert. Die Zustände eines Experiments (fertig geladen, Sitzung konfiguriert, Sitzung gestartet usw.) werden als Ereignisse an Module und *Panel* gemeldet, die wiederum ihre *Controllers* und *Boards* und diese wiederum ihre *Controls* benachrichtigen. Daten in einer Pipeline lösen ebenfalls ein Ereignis aus, das an den jeweils nächsten Filter bzw. alle Verwerter übermittelt wird.

4.2.3 Das Experiment

Für die erste Projektphase wird ein Simulator benötigt, der den Probanden über ein Bild eine Aufgabenstellung präsentiert. Ebenso soll es möglich sein, dass das simulierte System über einen eingblendeten Text eine Rückmeldung an die Versuchsteilnehmer gibt sowie eine Sprachausgabe übermitteln kann. Für den Wizard soll die Spracheingabe der Probanden hörbar, ein Videobild der Probanden sichtbar und die Möglichkeit, Rückmeldungen auszulösen, vorhanden sein.

In Abbildung 4a sieht man das Panel für den Wizard. Links unten ist ein Board platziert, das ein Video des Probanden und die bereits abgelaufene Zeit anzeigt. Links oben sieht man ein Board, mit dem die beiden Audiokanäle überwacht werden können. Das große Board rechts vereint drei Elemente. Unten ist eine Control eingelagert, mit der die Bilder ausgewählt werden können, die dann auf dem Panel für den Probanden, zu sehen in Abbildung 4b, erscheinen.⁷ Oben rechts sieht man eine schematische Darstellung des Probanden-Panels und oben links die Control, mit der die Rückmeldungen ausgelöst werden können.

Die vorgefertigten Rückmeldungen können für eine bessere Übersicht in Kategorien eingeteilt werden und bestehen neben einem Namen und einer Beschreibung aus zwei Zeichenketten und einer Audiodatei. Eine Zeichenkette wird als Text beim Probanden ausgegeben, die andere stellt die Transkription der Audiodatei dar und ist als Hilfestellung für den Wizard gedacht. Ist keine Audiodatei vorhanden, wird die Zeichenkette über eine Text-To-Speech Engine als Sprachausgabe an den Probanden gesendet. Die Zeichenketten können auch Variablen enthalten, die über ein weiteres Board, das nach der Auswahl einer Rückmeldung erscheint, über die

⁷Die integrierte Netzwerkfähigkeit von LCARSWT erlaubt Panel auf unterschiedlichen Rechnern.

Tastatur mit Werten belegt werden können. Die Textausgabe verschwindet automatisch nach einer festzulegenden Zeitdauer oder kann manuell über eine Schaltfläche entfernt werden. Dieses Verhalten wird über ein optionales Modul geregelt. Da beim Erstellen eines Experiments nicht an alle Eventualitäten gedacht werden kann, gibt es für den Wizard die Möglichkeit, direkt per Mikrofon zum Probanden zu sprechen. Damit zumindest klanglich alle Audioausgaben keine großen Unterschiede aufweisen, geht das Audiosignal vor der Übermittlung an den Probanden durch einen Vocoder. Dabei handelt es sich um eine externe Software der, am UCUI Projekt beteiligten, Firma Javox Solutions GmbH.

5 Zusammenfassung und Ausblick

Wir haben mit WoOF ein Framework begonnen, mit dem man ein flexibles Werkzeug zur Hand hat, um Systemsimulatoren für Wizard of Oz Experimente zu erstellen, die in ihrem Verhalten und ihrem Funktionsumfang erweiterbar sind. Dadurch eignet sich WoOF insbesondere für die nutzergetriebene Systementwicklung, bei der sinnvollerweise nacheinander mehrere Experimente durchgeführt werden, wobei der Simulator immer mehr Züge des fertigen Systems annimmt.

Für die weiteren Projektphasen werden wir WoOF durch die Einarbeitung der gemachten Erfahrungen und die Integration neuer Funktionalitäten verbessern. Module und Function-Provider sollen als optional markiert werden können und Objekte sollen die Möglichkeit bekommen, ihre Funktionen temporär abzulegen. Die Verarbeitung der Tastatur- und Touch-Eingaben soll an die Möglichkeiten von LCARSWT angepasst werden. Auch soll die Koppelung von mehreren Probanden-Panels integriert werden. Da die Probanden gefilmt werden, sollen sie sich nach Möglichkeit nicht zu viel bewegen, wenn sie per Touch-Eingabe mit dem System interagieren. Dies könnte durch ein zusätzliches Tablet erreicht werden, das den selben Bildschirminhalt darstellt wie der große Monitor, vor dem sie stehen, und die gleiche Interaktion wie mit dem Touch-Bildschirm ermöglicht. Die Basis hierfür ist bereits in WoOF enthalten.

Literatur

- [1] FRASER, N. M. ; GILBERT, G. N.: Simulating speech systems. In: *Computer Speech & Language* 5 (1991), Nr. 1, S. 81 – 99
- [2] GREEN, A. ; HÜTTENRAUCH, H. ; EKLUNDH, K. S.: Applying the Wizard-of-Oz framework to cooperative service discovery and configuration. In: *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2004, S. 575 – 580
- [3] KELLEY, J. F.: An Iterative Design Methodology for User-friendly Natural Language Office Information Applications. In: *ACM Trans. Inf. Syst.* 2 (1984), Jan., Nr. 1, S. 26 – 41
- [4] MANZESCHKE, A. ; WEBER, K. ; ROTHER, E. ; FANGERAU, H. : Ethische Fragen im Bereich Altersgerechter Assistenzsysteme / im Auftrag von VDI/VDE Innovation + Technik GmbH. Berlin, 2013. – Forschungsbericht
- [5] WATZLAWICK, P. ; BEAVIN, J. H. ; JACKSON, D. D.: *Menschliche Kommunikation*. 4., unveränd. Aufl. Bern [u.a.] : Huber, 1974
- [6] WEBER, K. : *Kommunikationswissenschaft*. Bd. 1: *Das Recht auf Informationszugang*. Berlin : Frank & Timme, 2005
- [7] WOLFF, M. : *LCARS Widget Toolkit*. <https://github.com/matthias-wolff/LCARSWT>. Version: 2.1.0-rc.1