

5 Umsetzung von Untersuchungsdesigns in Squeak

Knut Polkehn

Humboldt-Universität zu Berlin, Institut für Psychologie

Simulationen mit Squeak sind kein Selbstzweck. Im Forschungskontext dienen Sie dem Schaffen eines Untersuchungssettings, um eine bestimmte Fragestellung untersuchbar zu machen. Ein derartiges Vorgehen stellt zusätzliche Anforderungen an die Simulation, die sich in zwei Kategorien einteilen lassen. Neben Anforderungen, die sich erstens aus einem bestimmten Untersuchungsdesign ergeben, muss zweitens der spezifische Zugang zur Entwicklung einer Simulation in Squeak (z. B. Smalltalk, EToys, Seaside) berücksichtigt werden.

Als Beispiel soll in diesem Beitrag die Untersuchung einer empirischen Fragestellung verwendet werden. Zur Vereinfachung wird im Beispiel keine komplexe Simulation untersucht, sondern ein klassisches psychologisches Experiment – ein Stroop-Experiment – durchgeführt.

Im klassischen Stroop-Experiment werden die Probanden gebeten, die Farbe zu benennen, in der ein Wort abgebildet ist. Dabei handelt es sich um eine kongruente Bedingung, wenn das Wort seine Farbe selbst angibt (z. B. ist „grün“ in grün geschrieben). Bei einer inkongruenten Bedingung stimmt das Wort nicht mit seiner Farbe überein (z. B. ist „grün“ in rot geschrieben). Wenn es keine Ähnlichkeit zwischen der Bedeutung des Wortes und seiner Farbe gibt, nennt man die Bedingung „neutral“. Die Probanden benötigen bei Inkongruenz mehr Zeit zur Benennung der Farbe als bei einer neutralen oder einer kongruenten Bedingung (Stroop-Effekt).

5.1 Anforderungen durch das Untersuchungsdesign

5.1.1 Untersuchungsdesign

Zunächst muss man sich darüber klar werden, wie das Design der Untersuchung aussieht. Geht es um die Bewertung (Evaluation) von Prototypen oder werden beispielsweise in einem Experiment systematisch Bedingungen variiert, um den Einfluss der Manipulation von unabhängigen Variablen auf definierte abhängige Variablen zu untersuchen? Die Beantwortung derartiger Fragen hilft uns spezifische Anforderungen an die Gestaltung der Untersuchung zu formulieren. Wichtige Fragen sind:

- Werden ein oder mehrere Variablen variiert (manipuliert)?
- Welche Ausprägungen können diese Variablen haben?
- Werden die Probanden den Bedingungen zufällig zugewiesen (Experiment) oder anhand des zu variierten Merkmals ausgewählt (Quasi-Experiment)?
- Werden alle Probanden mit den verschiedenen Ausprägungen der unabhängigen Variable konfrontiert („within-Design“) oder wird jeweils eine Gruppe von Probanden einer bestimmten Ausprägung zugeordnet („between-Design“)?
- Müssen weitere Variablen erfasst werden (z. B. demographische Daten)?
- Muss die Verteilung von Probanden auf bestimmte Untersuchungsgruppen in Bezug auf weitere Merkmale kontrolliert werden?

Für die Umsetzung von Untersuchungsdesigns in Squeak bedeutet das folgendes:

- Probanden müssen unterschiedlichen Gruppen zugeordnet werden können.
- Diese Zuordnung muss nach verschiedenen Verfahren (zufällig, Gleichverteilung oder Balancierung nach einem dritten Merkmal, etc.) möglich sein.
- Es muss eine Steuerung des Versuchsablaufes möglich sein.

Im Beispiel soll die Variable „Kongruenz/Inkongruenz“ untersucht werden. Dabei sind drei Ausprägungen (kongruent, inkongruent, neutral) möglich. Die Probanden sollen allen verschiedenen Bedingungen wiederholt (within-Design), aber in zufälliger Reihenfolge ausgesetzt werden. Die Aufgabe der Probanden besteht darin, die Farbe des präsentierten Wortes durch Druck auf eine entsprechend kodierte Taste anzugeben.

5.1.2 Steuerung des Versuchablaufs

Nach der Betrachtung des Untersuchungsdesigns muss überlegt werden, wie der Ablauf der Untersuchung aussieht. Dazu gehört:

- die Zuweisung einer eindeutigen ID für den Probanden,
- das Umsetzen des eigentlichen Untersuchungsdesigns,
- das Darbieten von Instruktionen,
- das Darbieten von Vor- und Nachbefragungen,
- die Integration von Pausen.

Im Beispiel soll der Proband nach Start des Experiments automatisch eine eindeutige ID zugewiesen bekommen. An eine Instruktion soll sich eine Probephase anschließen, um zu sichern, dass der Proband die Instruktion befolgt. Danach folgt die eigentliche Untersuchungsphase. Diese soll aus vier Blöcken bestehen. Zwischen den Blöcken gibt es eine Minute Pause. Innerhalb eines Blockes werden den Probanden zufällig vier Begriffe je Kategorie vorgelegt. Vor jeder Präsentation eines Wortes soll zur Kontrolle der Aufmerksamkeitssteuerung eine Sekunde lang ein Fixierungskreuz dargeboten werden.

5.1.3 Messung der relevanten abhängigen Variablen

Zentral für die Untersuchung ist das Erfassen der abhängigen Variablen. Dazu ist zu bestimmen:

- Welche Daten sind relevante Daten? (abhängige Variablen und wichtige Kontextvariablen)
- Wie werden die relevanten Daten erfasst? (event- vs. zeitgetriggert oder in Kombination beider Verfahren)
- Wie kommen die Daten zum Experimentator / Untersuchungsleiter (bei computerbasierten Untersuchungen in der Regel durch das Schreiben von Logfiles auf dem Untersuchungsrechner)

Im Beispiel ist die Reaktionszeit der Probanden, also die Zeit von der Präsentation des Wortes bis zum Drücken einer Taste zur Angabe der Farbe dieses Wortes die relevante abhängige Variable. Gleichzeitig spielen aber auch noch weitere Kontextvariablen eine wichtige Rolle für die spätere Auswertung, wie die Korrektheit der Entscheidung, das präsentierte Wort und die Kategorie. Erfasst werden die Daten eventgetriggert, also ausgelöst durch den Start der Präsentation des Wortes bzw. den Tastendruck des Probanden. Die Daten müssen in einem Logfile abgelegt werden, welches sich in gängigen Datenbankanwendungen bzw. Statistikprogrammen weiterverarbeiten lässt.

5.1.4 Konfiguration durch den Untersuchungsleiter

Als letzter Punkt soll hier die Konfigurationsmöglichkeiten durch den Untersuchungsleiter erwähnt werden:

- Bestimmen der Art und Weise des Vergebens der Probanden-ID (Code, fortlaufend Nummer, ...)
- Konfigurieren von wichtigen Parametern der Untersuchung
- Auswahl von Untersuchungsmaterialien
- Bestimmen von Verzeichnissen für das Speichern von Logfiles

Im Beispiel soll die die Vergabe von Probanden-IDs sowohl per Code als auch mit fortlaufender Nummerierung möglich sein. Als wichtige Parameter sollen für spätere weitere Untersuchungen die Pausenzeiten sowie die Erscheinungsdauer des Fixierungskreuzes einstellbar sein. Weiterhin soll es möglich sein unterschiedliche Wortlisten zu verwenden.

5.2 Anforderungen aus dem Entwicklungszugang

Squeak lässt verschiedene Zugänge zur Entwicklung von Simulationen zu. Was bedeutet das für die Umsetzung von Untersuchungsdesigns?

5.2.1 Entwickeln mit EToys

Die Entwicklung von Anwendungen mit Squeak-EToys hat zunächst viele Vorteile. Das Erzeugen und Manipulieren multimedialer Objekte und deren Animation über Skripting per Maus lassen das schnelle Erstellen auch komplexerer Anwendungen zu, selbst wenn der Entwickler über keinerlei Smalltalk-Kenntnisse verfügt. Eine fertige EToy-Anwendung kann als Projekt exportiert werden und über das Browser-Plugin bzw. über ein normales Squeak-Image wieder importiert werden. Zu beachten ist allerdings, dass man immer mit dem Prototypen arbeitet – eine Generalisierung durch Klassenbildung erfolgt zunächst nicht. Man kann jedoch einzelne Bestandteile des Projektes (Buttons, Instruktionen) zur späteren Wiederverwendung exportieren, solange keine projektinternen Abhängigkeiten existieren.

Für das Beispiel muss eine zufällige Zuweisung eines Wortes aus einer Liste von Begriffen möglich sein. Abbildung 11 zeigt eine einfache Realisierung der Präsentation des Materials. Mit dem Abfragen des keystroke-Ereignisses lässt sich leicht ein eventgetriggertes Loggen verwirklichen. Ein Umschalten des Skript-Auslöser-Events auf „ticker“ würde ein zeitgetriggerte Abarbeiten von „Script1“ ermöglichen.

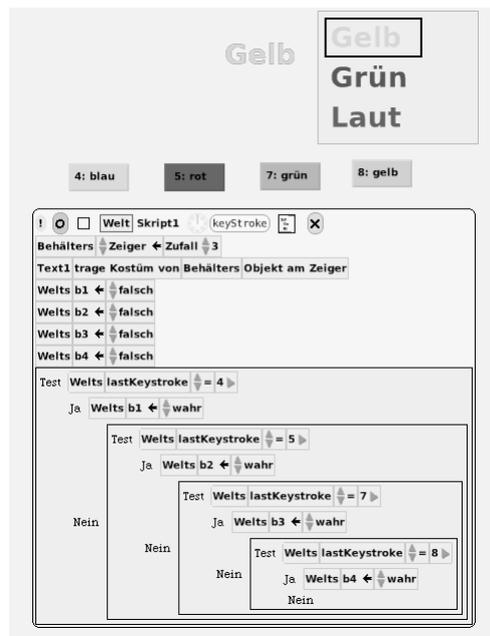


Abbildung 11: Entwicklung eines Stroop-Experimentes mit EToys

Schwieriger (und da liegt ein entscheidender Nachteil) ist es, aus den EToys heraus auf Reaktionszeiten zuzugreifen (im Beispiel die zentrale abhängige Variable). Das lässt sich hier nur verwirklichen, indem die Scripte in den Smalltalk-Modus umgeschaltet und durch entsprechende Syntax angereichert werden. Für die komplette Versuchssteuerung, die Ausgabe externer Daten sowie die Konfiguration wären noch entsprechende Tools zu entwickeln – idealerweise in Form von EToy-Versuchssteuerungs-Objekten.

5.2.2 Entwickeln mit Smalltalk

Mit den üblichen Smalltalkwerkzeugen (Klassenbrowser, workspace, ...) kann ein erfahrener Smalltalk-Entwickler aus dem Vollen schöpfen. Größter Vorteil ist die Möglichkeit, die EToys bzw. die dahinter liegende Morph-Klassenbibliothek für die objektorientierte Entwicklung mit Smalltalk zu nutzen. Durch das einfache Zuweisen von

event-handlern bzw. durch Nutzung der standardisierten Stepping-Methode ist das event- bzw. zeitgetriggertes Aufzeichnen von Ereignissen unproblematisch möglich.

Für das Beispiel müsste eine Stroop-Klasse programmiert werden würde, die idealerweise parametrisierbar für die Wiederverwendbarkeit in verschiedenen Untersuchungskontexten ausgelegt wäre. Alle Anforderungen aus der Betrachtung des Untersuchungsdesigns sind mit Smalltalk prinzipiell umsetzbar, erfordern jedoch eine entsprechende Vorerfahrung. Eine Klassenbibliothek, die Versuchssteuerungsobjekte und Methoden für die wichtigsten Aspekte des Untersuchungsdesign beinhaltet, wäre außerordentlich hilfreich.

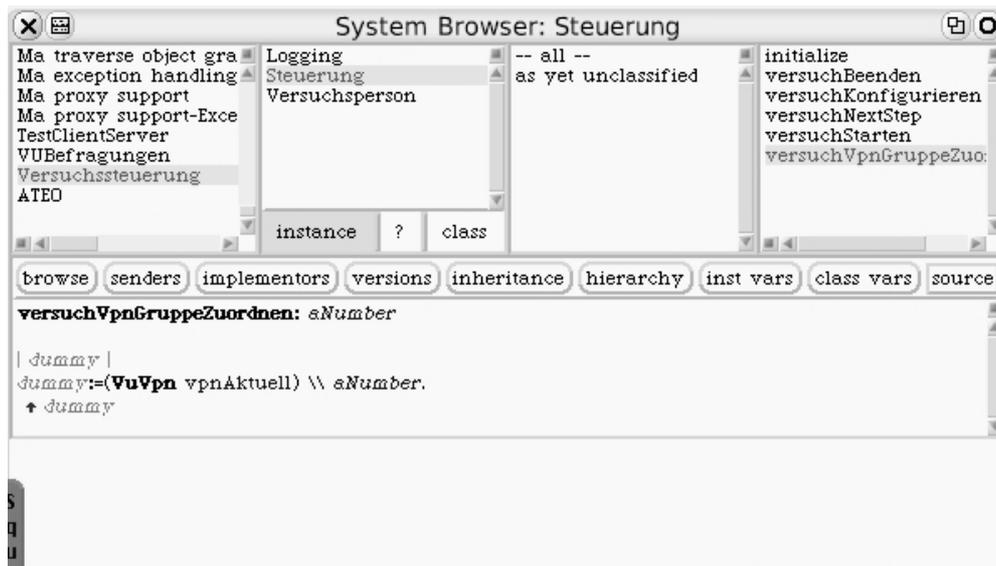


Abbildung 12: Entwicklung von Versuchssteuerungsobjekten mit Smalltalk

Ein letzter Punkt soll nicht unerwähnt bleiben. Bottom-Up- (EToys) und Top-Down-Entwicklung (Smalltalk) sollten sich ergänzen. Dafür wäre es notwendig, dass über EToy-Prototypen auf Klassen generalisiert werden kann. Diese Möglichkeit existiert bisher in Squeak nicht.

5.2.3 Seaside: Entwickeln für das WWW

Seaside (www.seaside.st) ist ein Webframework für Smalltalk und ist für Squeak verfügbar. Für die Umsetzung von Untersuchungsdesigns ist Seaside aus verschiedenen Gründen interessant:

- Seaside ist ein Framework zum Erstellen dynamischer Webanwendungen. Damit ist es prinzipiell für die Durchführung von Web-Experimenten geeignet. Für Squeak-EToys existiert zwar ein Browser-Plugin, bei dessen Nutzung man sich aber der Nachteile (z. B. notwendige Installation beim Client, Probleme beim Übermitteln von Logfile-Daten an den Untersuchungsleiter) bewusst sein muss. Seaside-Anwendungen lassen sich browser-unabhängig gestalten.
- In Seaside wird konsequente Objektorientierung mit einem so genannten „continuation-based framework“ kombiniert. Eine Web-Applikation besteht aus verschiedenen Komponenten mit einem jeweils eigenen Control-Mechanismus. Jede dieser Komponenten kann auch mehrfach auf einer Seite vorhanden sein. Ein spezieller Call-back-Mechanismus regelt den Umgang mit Links und Formular-Aktionen.
- Seaside liegt ein komplettes Smalltalk-System zugrunde. Die damit einhergehende „Liveness“ des Smalltalksystems ermöglicht ein vollständiges Sessionmanagement und vor allem die Lösung des Back-Button-Problems (fehlende Passung Browserzustand zu Zustand des Web-Servers bei dynamischen Anwendungen). Das Betätigen des Back-Buttons kann bei Aktivierung des Session-Back-Trackings in Seaside tatsächlich Nutzeraktionen rückgängig machen.

- Interessant für das Durchführen von Untersuchungen ist die Möglichkeit der Wiederverwendung von in Smalltalk implementierten Modellen (Webapplikation als anderer View für dieselben Modelle).
- Seaside unterstützt die Nutzung von Stylesheets und Javascript, und aktuell auch AJAX, was die Entwicklung dynamischer Anwendungen ohne Neuladen von Seiten ermöglicht.
- Die Entwicklung von Seaside-Applikationen erfolgt wie die Entwicklung von Smalltalk-Anwendungen (inkrementelles Entwickeln im laufenden/lebenden System) mit allen Debugging-Möglichkeiten (vgl. Abbildung 13). Unter Nutzung einer speziellen Toolbar kann die Entwicklung selbst gleichfalls webbasiert erfolgen.

Das Beispiel lässt sich mit Seaside sehr gut unter Nutzung eines vorhandenen Smalltalk-Stroop-Experiments umsetzen. Neu ist die Ausgabe des Begriffs in ein vorhandenes DIV der HTML-Seite unter Nutzung von AJAX-Mechanismen, die Abfrage der gedrückten Taste sowie die Messung von Zeiten im Client mit Javascript (die Zeitmessung darf wegen der variablen Systemantwortzeiten im Internet nicht auf dem Server erfolgen).

Seaside ist für Web-Untersuchungen geeignet, die wenig Dynamik im Client verlangen. So sind die Tracking-Experimente in der ATEO-Mikrowelt (siehe Beitrag Groß) nicht als Webanwendung umsetzbar. Für die Simulation von Geräten, wie bspw. Handy oder MP3-Player (siehe Beitrag Bruder et al.) scheint die Nutzung von Seaside ein vielversprechender Ansatz zu sein.

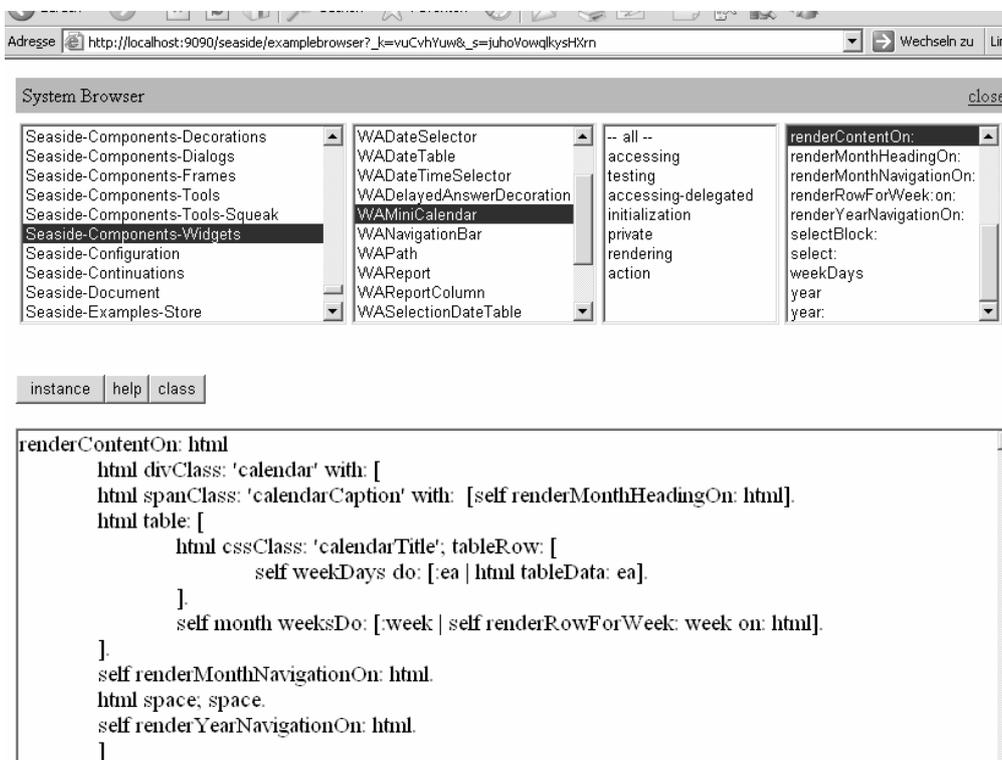


Abbildung 13: Webbasierte Entwicklung mit Seaside am Beispiel eines Kalenders

5.3 Fazit

Die Nutzung von Squeak für die Umsetzung von Untersuchungsdesigns scheint ein vielversprechender Ansatz zu sein. Wichtig für eine effiziente Umsetzung über alle drei Zugänge (EToys, Smalltalk, Seaside) ist die Entwicklung einer Klassenbibliothek für Versuche, die Klassen mit den entsprechenden Methoden für die Versuchsperson, für Gruppen, die eigentliche Steuerung und das Generieren von Logfiles enthält.

Eine empirische Untersuchung kann dann auf den verschiedenen beschriebenen Wegen implementiert werden. Für den Zugang über EToys ist es wichtig, auch die Versuchssteuerungsobjekte aus den EToys heraus im Zugriff zu haben (z. B. über Smalltalk im Script oder eine Versuchs-Flap mit VU-Objekten, ...).

Squeak besitzt ein großes Potential, aber eine Entwicklung von professionellen Hilfsmitteln für die verschiedenen Nutzungsszenarien in Squeak ist notwendig.

Kontaktinformationen

Humboldt-Universität zu Berlin
Institut für Psychologie
Abteilung für Ingenieurpsychologie/Kognitive Ergonomie
Knut Polkehn

Rudower Chaussee 18
12489 Berlin
knut.polkehn@rz.hu-berlin.de

Tel.: + 49 30/2093-9357
Fax.: +49 30/2093-2021 9357