M. Koch, A. Butz & J. Schlichter (Hrsg.): Mensch und Computer 2014 Tagungsband,
München: Oldenbourg Wissenschaftsverlag, 2014, S. 295-298.

Auswertung von Reverse Card Sorting Experimenten

Daniel Brumberg, Gerd Szwillus

Institut für Informatik, Universität Paderborn

Zusammenfassung

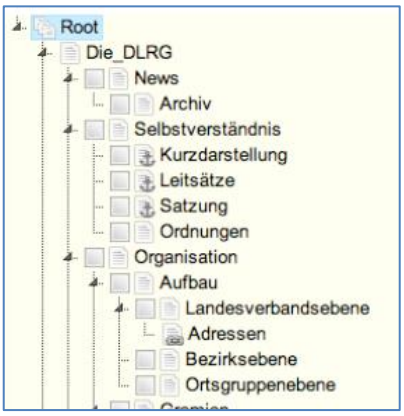
In Reverse Card Sorting (RCS) Experimenten wird untersucht, inwieweit die Benennung und Strukturierung der Begriffe der Informationsarchitektur eines Webauftritts für die Besucher plausibel und nachvollziehbar definiert wurde. Im Gegensatz zum Usability-Test an den fertig gestalteten Webseiten orientiert sich das RCS-Experiment ausschließlich an den verwendeten Linkankertexten und blendet damit einen möglichen positiven oder negativen Einfluss durch die visuelle Gestaltung aus. In diesem Paper wird eine softwareunterstützte Methode zur Auswertung derartiger Experimente beschrieben und an einem realistischen Beispiel demonstriert. Die Methode erlaubt darüber hinaus eine Kategorisierung typischen Fehlverhaltens des Besuchers während der Navigation.

1 Reverse Card Sorting

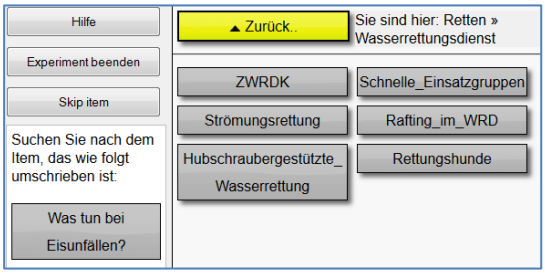
Es passiert nicht selten, dass man während der Navigation innerhalb eines Webauftritts plötzlich vor der Frage steht: „Wie geht es denn hier weiter?“ Man hat ein bestimmtes Ziel, möchte eine konkrete Fragestellung klären, ist sich auch sicher, dass die Information in dem Webauftritt vorhanden ist, findet aber im sichtbaren Angebot an Links keinen, der richtig oder auch nur einigermaßen passend erscheint. Um solche Situationen im Vorfeld zu erkennen, kann der Entwickler des Auftritts Usability-Tests durchführen – allerdings muss dafür der Webauftritt bereits existieren. Eine Alternative stellt Reverse Card Sorting (RCS) dar.

Um RCS-Experimente durchzuführen, wird nicht der fertige Webauftritt benötigt, sondern lediglich eine Spezifikation der geplanten Informationsarchitektur (IA). Diese wird als Navigationsstruktur in das Verfahren eingebracht, die die Namen der Links und die Verlinkungsstruktur beinhaltet. Die gemeinhin als Hierarchie bezeichnete Navigationsstruktur ist bei realen Webauftritten in den seltensten Fällen eine "echte" Hierarchie, sondern typischerweise eine zusätzlich nicht-hierarchisch vernetzte Hypertextstruktur, allerdings mit einer unterliegenden Hierarchie; sie wird daher vom Besucher trotzdem als primär hierarchisch empfunden. Für die hier behandelten RCS-Experimente wurde die Webapplikation WeCaSo (Vdovkin 2009; Barbula 2011) verwendet; die Abbildung auf der nächsten Seite zeigt einen Ausschnitt der Spezifikation der Navigationsstruktur der Website der DLRG, die als Fallstudie untersucht wurde. Die unterliegende Hierarchie wird darin deutlich, allerdings sind auch

viele Verbindungen „quer“ zur Hierarchie eingefügt worden (erkennbar am „Link“-Symbol, etwa bei „Adressen“).



Basierend auf der Spezifikation der Navigationsstruktur werden in RCS-Experimenten den Versuchspersonen Navigationsaufgaben gestellt, die sie in der spezifizierten Informationsarchitektur durch Auffinden der „richtigen“ Zielseite bewältigen müssen. Eine Beispielaufgabe aus dem DLRG-Experiment etwa war die Suche nach Informationen zur Frage „Was tun bei Eisunfällen?“ Die folgende Abbildung zeigt die Sicht der Versuchspersonen im Laufe des Experimentes bei Bearbeiten dieser Aufgabe.



Aus der Literatur sind mehrere RCS-Systeme bekannt, wie etwa TreeJack (Optimal Workshop 2012) oder als Komponente von MindCanvas (Uzanto Consulting) bzw. der C-Inspector (Schilb). Allerdings ist allen Varianten gemeinsam, dass sie von echten Hierarchien ausgehen, was eine unrealistische Voraussetzung ist. Gerade das Zulassen nicht-hierarchischer „Querverbindungen“ gestaltet die Analyse schwierig.

2 Auswertung

Während des Experiments zeichnet WeCaSo die durchlaufenen Knoten, gewählten Links, und auch die zugehörigen Wartezeiten auf; auch ein eventueller Abbruch wird dokumentiert. Als Ergebnis eines WeCaSo-Experimentes entsteht daher pro Aufgabe ein Pfad der durchlaufenen Knoten mit zugeordneten Zeitangaben. Da der Webauftritt nicht streng hierarchisch ist, kann es durchaus mehrere, verschiedene Pfade geben, die zum Ziel führen.

Um systematisch an die Auswertung heranzugehen, wird aufbauend auf einem Grundgerüst an Definitionen festgelegt, wie „weit“ zwei Seiten in der IA voneinander entfernt sind (Distanzbegriff). Diese Definition erlaubt es, Problemstellen in einem Pfad zu erkennen, indem wir untersuchen, ob der Besucher sich in einem Navigationsschritt vom Zielknoten entfernt, statt sich ihm zu nähern, oder auch ob er zwar den richtigen Weg geht, darüber aber sehr lange nachdenkt. Um einen gesamten Pfad in seiner „Falschheit“ zu bewerten, muss berücksichtigt werden, dass der Pfad mehrere solcher Problemstellen enthalten kann, und dass „frühe“ Fehler bei der Navigation als stärker zu bewerten sind, da sie den Besucher in völlig falsche Bereiche der IA führen. Hierzu werden die Knoten in der Navigationsstruktur mit einem sogenannten Rang bewertet, der den Abstand zum Startpunkt repräsentiert. Zusätzlich wird die Wartezeit vor der nächsten Entscheidung einbezogen, wobei wir davon ausgegangen sind, dass lange Wartezeiten gleichzusetzen sind mit einer schwierigen Entscheidungssi-

tuation für die Versuchsperson. Insgesamt entstand so ein Kalkül, mit dem durch Analyse aller gesammelten Pfade der Versuchspersonen eine Bewertung der einzelnen Seiten des Webauftritts als mehr oder weniger problematisch berechnet werden kann (Brumberg 2013).

Bei der Betrachtung der Experimentergebnisse aus der DLRG-Fallstudie kristallisierten sich typische Verhaltensstrukturen der Versuchspersonen heraus. Wir haben diese Fehlverhaltensmuster gezielt untersucht, um sicherzustellen, dass der oben skizzierte Analysealgorithmus die Muster zuverlässig erkennt. Für die folgenden vier Muster war das der Fall: Durchlaufen eines direkten (ohne Kreise), aber unnötig langen Pfades zum Ziel, Durchlaufen eines Kreises, Wiederholen einer bereits als Fehler erkannten Entscheidung und Abbruch oder Umkehren, obwohl auf dem richtigen Weg.

Ein letztes gefundenes Fehlverhaltensmuster, „systematisches Suchen“ genannt, erwies sich allerdings als sehr problematisch. Dabei hat der Besucher es aufgegeben, über die Bedeutung der Links nachzudenken, sondern probiert (genervt!) systematisch alle Links der Reihe nach durch. Es wäre verfehlt, dieses Verhalten analog zu gezieltem Verhalten zu untersuchen. Bei diesem Verhalten treten eine Häufung von Problemstellen, Klicks und „Zurück“-Klicks, sowie kurze Verweildauern auf. Zusammen mit einem Maß für die nötige „Lesezeit“ konnten wir schließlich eine parametrisierte Heuristik aufstellen, die dieses Fehlverhaltensmuster in den meisten Fällen erkennt und damit eine Verfälschung der Analyseergebnisse verhindert.

3 Fallstudie

Als Fallstudie wurde eine existierende Website (www.dlrg.de) in WeCaSo nachgebaut, Stand: 19.02.2013. Zu diesem Zeitpunkt bestand der Auftritt aus 543 Unterseiten mit 646 internen Links, sodass ein realistisch großes Versuchsobjekt vorliegt. Es wurden insgesamt 15 Navigationsaufgaben definiert, die so ausgewählt wurden, dass sehr verschiedene und auch verschieden tiefe Bereiche des Auftritts besucht werden mussten. Insgesamt haben 50 Versuchspersonen das RCS-Experiment durchgeführt. Als globales Ergebnis wurde ermittelt, dass die Lösungsquote bei 74% lag, wobei es große Unterschiede zwischen den einzelnen Aufgaben gab (s. Abbildung oben). So wurde die Aufgabe „Unfallversicherung Taucher“ von über 73% der Teilnehmer, "Einsatztauchen" aber nur von knapp 44% der Teilnehmer gelöst. Weitere Tabellen zeigen, auf welchen Seiten die Testpersonen „zu lange“ überlegt haben oder vermehrt falsch "abgebogen" sind. Auch Informationen dazu (wie auf der

Aufgaben-Übersicht:

Aufgabe	Lösungsquote
Einsatztauchen	43,48 Prozent
Selbstrettung_bei_Eisunfällen	52,27 Prozent
Realistische_Unfall-_und_Notfalldarstellung	54,55 Prozent
nach_Bundesländern	65,12 Prozent
Strömungsrettung	71,11 Prozent
Klaus_Wilkens	72,73 Prozent
Unfallsversicherung_Taucher	73,33 Prozent
Bubnen und Walzen	75,56 Prozent

Details anzeigen:

nach Problem-Seite: Startseite

nach Aufgabe: Wiedervereinigung

Details zu Problem-Seite "Startseite"

Aufgabe	Problem-Verteilung	Latenz in ms
Einsatztauchen	13,03 Prozent	3144
Selbstrettung_bei_Eisunfällen	7,78 Prozent	9229
Realistische_Unfall-_und_Notfalldarste...	8,90 Prozent	8694
nach_Bundesländern	9,21 Prozent	4782
Strömungsrettung	5,33 Prozent	6013

vorigen Seite gezeigt), inwieweit eine bestimmte Seite zu bestimmtem Fehlverhalten beigetragen hat, kann abgerufen werden. Hier sieht man etwa, dass die Startseite in rund 13% der Fälle zum Nichtauffinden der Seite „Einsatztauchen“ beigetragen hat.

4 Schlussbemerkungen

Das vorliegende Paper zeigt die Möglichkeiten der Auswertung von RCS-Experimenten anhand einer realistisch großen Fallstudie auf. Das Verfahren wurde softwaremäßig implementiert und mit einer signifikant großen Zahl von Versuchspersonen erprobt. Es ist allerdings nur ein erster Schritt, da viele Fragen noch offen bleiben. Zum einen ist es eine Kernfrage zur Rechtfertigung des gesamten Ansatzes, ob mit RCS die gleichen Probleme gefunden werden, wie mit "echten" Usability-Tests am realen Webauftritt. Die visuelle Gestaltung der Webseite beeinflusst entscheidend die Wahrnehmung aller Elemente, auch der Links, und damit den Navigationserfolg. Die Repräsentation aller Links als neutral gestalteter, gleich aussehender Buttons ist demgegenüber unnatürlich. Auch die Tatsache, dass die Versuchspersonen streng ebenenweise durch die Hierarchie geführt werden, entspricht nicht der gängigen Praxis in realen Webauftritten. Die Methode hat aber den großen Vorteil, dass die IA viel früher getestet und dann auch modifiziert werden kann, und es gibt deutliche Hinweise, dass es eine erhebliche Überschneidung gefundener Probleme zwischen RCS- und Usability-Test gibt (Zhao 2013).

Literaturverzeichnis

Barbula, A. (2011). Integration hierarchischer Sortierung, Multiple-Insert und Reverse-Card-Sorting in eine klassische Card-Sorting-Applikation. Bachelorarbeit. Institut für Informatik. Paderborn.

Brumberg, D. (2013). Entwicklung einer Java-Anwendung zur Auswertung von Reverse-Card-Sorting-Versuchen. Bachelorarbeit. Institut für Informatik. Paderborn.

Optimal Workshop (2012): Treejack. Information Architecture Validation Software. Online verfügbar unter http://www.optimalworkshop.com/treejack.htm, zuletzt geprüft am 6.3.13.

Schilb, S.: C-Inspector. A tool for information architecture testing. Steffen Schilb, Köln. Online verfügbar unter http://www.c-inspector.com/index.php, zuletzt geprüft am 6.3.13.

Uzanto Consulting (Hrsg.): Game-like Elicitation Methods. Online verfügbar unter http://www.themindcanvas.com.

Vdovkin, A. (2009). Entwicklung einer webbasierten Card-Sorting-Applikation. Bachelorarbeit. Institut für Informatik. Paderborn.

Zhao, H. (2013). Vergleichende Untersuchung von Navigationshierarchien mit Reverse Card Sorting und Benutzertests. Masterarbeit. Institut für Informatik. Paderborn.