

Transfer komplexer Interaktionen von Desktop- zu Web-Anwendungen

Alexander Nolte, Michael Prilla

Informations- und Technikmanagement, Ruhr-Universität Bochum

Zusammenfassung

Im Zuge der immer stärkeren Ausbreitung von Anwendungen und Technologien im Bereich des Web 2.0 stellt sich die Frage, wie der Transfer komplexer Interaktionen von Desktop- zu Web-Anwendungen zu bewerkstelligen ist. Für die dazu notwendige Anpassung des Funktionsumfangs erarbeitet die vorliegende Analyse eine systematische Vorgehensweise am Beispiel eines Editors für Prozessdiagramme. Weiterhin werden Gestaltungshinweise abgeleitet, die sowohl die Erwartungen der Nutzer an die Funktionsweise einer Desktop-Anwendung als auch an das Medium Internet berücksichtigen. Die einschlägige Literatur beschäftigt sich häufig lediglich mit der Fragestellung, wie die Usability einer Webseite von den im Web 2.0 üblichen Möglichkeiten zur individuellen Gestaltung der Oberfläche beeinflusst wird. Die speziellen Erfordernisse bei der Übertragung komplexer Interaktionen von einer Desktop- in eine Web-Anwendung werden aus unserer Sicht nicht hinreichend betrachtet.

1 Einleitung

In den letzten Jahren haben sich Anwendungen und Technologien des Web 2.0 zur Kommunikation und Kooperation zwischen Nutzern des Internets etabliert. Neben den häufig genannten Vertretern des Web 2.0 (O'Reilly 2007) wie Wikis, Blogs oder Social Bookmarking werden zunehmend interaktive Anwendungen angeboten, die ursprünglich aus dem Desktop-Bereich stammen. Solche so genannten **Applications on the Web (AoW)** wie bspw. Google Docs & Spreadsheets¹ bieten Nutzern in einem Webbrowser umfangreiche Funktionen, die bis dato Desktop-Anwendungen vorbehalten waren, wobei Formen der textbasierten Bearbeitung im Vordergrund stehen. Grafikeditoren stellen zusätzlich komplexere Anforderungen aufgrund der nicht-linearen Anordnung von Elementen.

In der heutigen Diskussion des Web 2.0 stellen AoW im Vergleich zur Aufmerksamkeit gegenüber populäreren Vertretern wie Wikis und Blogs noch immer Randerscheinungen dar.

¹ <http://docs.google.com>

Gleichwohl sind sie typische Beispiele der mit dem Web 2.0 oftmals verbundenen Potentiale. So lösen sie die Trennung zwischen Konsumenten und Produzenten von Inhalten auf, erschließen neue Nutzergruppen für Anwendungstypen und ermöglichen eine Geräte- und statusunabhängige Zusammenarbeit von Akteuren über das Internet. Die Verbindung dieser Charakteristika mit Interaktionen, die Nutzer von Desktop-Anwendungen kennen, und die Überführung vorhandener Funktionalität in AoW sind daher Schlüssel zu einer erfolgreichen Nutzung interaktiver Anwendungen im Web.

Die Überführung von Anwendungen, die Nutzer aus dem Desktop-Bereich gewöhnt sind, in Anwendungen, die in einem Webbrowser bedient werden können, stellt eine Herausforderung an die Interaktions- und Oberflächengestaltung dar. Dies liegt nur zum Teil an den unterschiedlichen technischen Möglichkeiten von Desktop- und Web-Interaktionen. Die von uns durchgeführte Untersuchung hat gezeigt, dass auch die unterschiedlich gewachsenen Erwartungshaltungen den Transfer erschweren. AoW nehmen nach unseren Erkenntnissen in ihrer Umsetzung hier eine Position zwischen den Stühlen ein: Grundlegende Interaktionsmuster bspw. einer Textverarbeitung basieren auf Prinzipien der Desktop-Interaktion, müssen aber unter den Voraussetzungen und entsprechend der Erwartungshaltung von Nutzern im Web gestaltet werden. Gleichwohl finden sich derzeit keine uns bekannten Beiträge, die sich mit dem Transfer einer Desktop- in eine Web-Anwendung beschäftigen. Das Vorgehen bei einer solchen Aufgabe scheint also eher pragmatischen Rationalen zu unterliegen. Dies drückt sich zusätzlich in der voreiligen Annahme aus, dass AoW es auch ungeübten Nutzern erlauben sollen, komplexe Interaktion ohne vorherige Schulung auszuführen (McAfee 2006). Wir argumentieren, dass dazu eine systematische Reduktion des Funktionsumfangs sowie die Berücksichtigung der unterschiedlichen Erwartungshaltungen an eine Desktop-Anwendung im Vergleich zum Medium Internet erforderlich sind. Dieser Beitrag beschreibt das Vorgehen bei der Identifikation des benötigten Funktionsumfangs und unternimmt den Versuch Kriterien zu finden und Vorschläge zu entwickeln, den Problemen bezüglich der unterschiedlichen Erwartungshaltungen zu begegnen. Hierbei wird keine umfassende Transfermethode entwickelt – eine solche kann lediglich durch einen breit angelegten Vergleich mit zu diesem Beitrag ähnlichen Studien erstellt werden. Gleichwohl stellt der Beitrag Merkmale eines erfolgreichen Transfers vom Desktop in das Web 2.0 dar und kann daher als Ausgangsbasis für weitere Forschungs- und Entwicklungsvorhaben dienen.

Den Vorteil einer AoW gegenüber einer netzwerkfähigen Desktop-Anwendung sehen wir speziell in deren Mobilität. So ist es mit Hilfe einer AoW – unter Berücksichtigung einer entsprechenden technischen Umsetzung – möglich, jederzeit auf jedem browserfähigen Endgerät mit Zugang zum Internet deren Funktionen zu nutzen. Damit kann zusätzlich eine wesentliche Barriere zur Nutzung von Prozessmodellen abgebaut werden, welche – laut unserer Erfahrung in Forschung und Lehre – unter Anderem in der Verfügbarkeit von Tools bzw. Lizenzen für diese besteht. In dem konkreten Fall der Umsetzung eines Editors für Prozessdiagramme in eine AoW sehen wir insbesondere den Vorteil, mittels einer solchen AoW das kollaborative Modellieren unterstützen zu können. Dieses kann zeitlich wie räumlich synchron und asynchron erfolgen, sofern ein zuverlässiger Mechanismus zur Vermeidung von Synchronisationskonflikten vorgesehen ist.

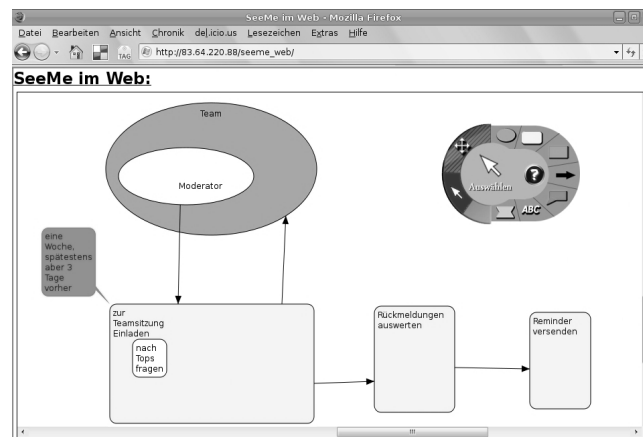


Abbildung 1: Modell innerhalb des SeeMe Webeditors (Cockpit rechts oben)

In den folgenden Abschnitten beschreiben wir zunächst allgemeine Gestaltungsprinzipien für Desktop- und Web-Anwendungen, und zeigen die unterschiedlichen Erwartungshaltungen an die Interaktion. Anschließend stellen wir Durchführung und Ergebnisse einer Fallstudie vor, deren Ziel es war, den Prototyp eines Webeditors zur Prozessmodellierung zu entwickeln und anschließend zu evaluieren. Dabei standen die Reduktion des Funktionsumfangs sowie deren gebrauchstaugliche Umsetzung im Vordergrund. Der darauf folgende Abschnitt vergleicht die gewonnenen Erkenntnisse mit der Umsetzung bestehender AoW und ordnet sie in den Kontext der aktuellen Forschung ein. Der Beitrag endet mit einer Zusammenfassung sowie einem Ausblick auf weitere Arbeiten im Bereich der Gebrauchstauglichkeit von AoW.

2 Anforderungen an die Usability von AoW

Es existieren verschiedene Kriterienkataloge zur Beurteilung der Usability von Desktop Anwendungen. Dazu gehören neben der eher methodischen ISO Richtlinie 9241 Teil 11 (DIN EN ISO 9241-11 1998) die entsprechenden „Grundsätze der Dialoggestaltung“ (DIN EN ISO 9241-110 2006) und weitere Heuristiken (Nielsen 1994; Cockton et al. 2003). Alle beschreiben einheitlich die Erfüllung der Nutzererwartungen an die Anwendung als entscheidendes Kriterium. Darunter ist speziell die Möglichkeit der Nutzer zu verstehen, sich softwareübergreifend auf allgemein gültige Standards zu verlassen. Dazu zählt beispielsweise das Öffnen eines Kontextmenüs zu dem ausgewählten Bestandteil eines Dokuments bei Betätigung der rechten Maustaste.

Im Hinblick auf die Web-Usability fällt auf, dass das Design einer leicht verständlichen Navigation sowie die Umsetzung einer guten Suchfunktion im Zentrum des Interesses stehen (Galitz 2002). Dies ist der Tatsache geschuldet, dass Webseiten lediglich zur Darstellung und Verbreitung von Informationen dienen. Somit bestanden die einzigen zu designenden Ab-

läufe innerhalb einer Webseite in der Zeit vor Web 2.0 in der effektiven und effizienten Exploration des Inhaltes mit Ausnahme von Funktionalitäten zum Ausfüllen von Formularen.

Zusätzlich spielt die Einsteiger-Freundlichkeit bei einer Webseite im Vergleich zu einer Desktop-Anwendung eine noch zentralere Rolle. Ist ein Nutzer nach der Installation einer Desktop Anwendung bereit, Zeit zu investieren, um deren Benutzung zu erlernen, fällt diese Zeit bei der Verwendung einer Webseite deutlich geringer aus, da die Nutzung ähnlicher Angebote im Web vergleichsweise leicht fällt (Nielsen 2000). Aus diesem Grund ist es erforderlich, die AoW so zu gestalten, dass ihre Nutzung sich selbst Einsteigern intuitiv erschließt. Somit ist im Hinblick auf ihre Umsetzung eine Reduktion des Funktionsumfangs erforderlich. Diese muss – wie in unserer Analyse dargestellt – systematisch bezogen auf den späteren Anwendungskontext erfolgen.

Auch die Erfüllung der Nutzererwartung erweist sich als nicht trivial, da die Erwartungen an die Interaktion mit einer Webseite sich von denen an eine Desktop Anwendung unterscheiden. Während beispielsweise das Öffnen eines verlinkten Inhaltes innerhalb einer Desktop-Anwendung durch einen Doppel-Klick erfolgt, bewirkt bei einer Webseite bereits ein einfacher Klick die entsprechende Aktion.

3 Fallstudie: Transfer eines Desktop-Editors für Prozessmodellierung in eine AoW

Unsere Untersuchung beschäftigt sich mit dem Transfer des SeeMe Desktop Editors² für die Modellierungsmethode SeeMe (Herrmann 2009) in eine AoW (siehe Abbildung 1). Der Analyse der Anforderungen liegt unserer Untersuchung folgendes Szenario zu Grunde:

Innerhalb einer Arbeitsgruppe, deren Mitglieder über Basiswissen bezüglich der Modellierungsmethode SeeMe verfügen, möchten die Mitglieder Aspekte eines Prozesses skizzieren. Diese sollen später zusammengeführt und gemeinsam weiterentwickelt werden. Dazu verwendet man den in Abbildung 1 dargestellten Webeditor. Anschließend gibt man die entsprechenden URLs an einen Moderator der Arbeitsgruppe weiter. Dieser fügt die Modell-Fragmente zusammen. Abschließend konsolidiert man den Entwurf in einem Workshop, um ein gemeinsames Modell zu erhalten – hierzu verwendet man eher einen Desktop-basierten Modell-Editor, der auch Funktionen enthält, mit denen man eine Ästhetisierung des Modells vornehmen kann – diese spielen in der Skizzierungsphase mit dem Web-Editor keine Rolle.

Zunächst wurde eine erste Erhebung durchgeführt, welche insbesondere die Identifikation der innerhalb des Szenarios benötigten Funktionen und somit eine Reduktion des Funktionsumfang zum Ziel hatte. In einer zweiten Phase wurde ein erster Prototyp hinsichtlich des in der ersten Phase festgestellten Funktionsumfangs evaluiert und anhand der gewonnenen Erkenntnisse weiterentwickelt. Die Ergebnisse der Untersuchung wurden weiterhin genutzt,

² <http://www.imtm-iaw.rub.de/projekte/seeme/index.html>

um eine gebrauchstaugliche Umsetzung speziell hinsichtlich der Einsteiger-Freundlichkeit zu ermöglichen. Dazu konnten Empfehlungen entwickelt werden, die sich allgemein auf die Optimierung der Einsteiger-Freundlichkeit bei AoW anwenden lassen.

Die folgenden Abschnitte beschreiben Aufbau und Verlauf der Studie, sowie die daraus abgeleiteten Erkenntnisse. Besonderes Augenmerk liegt dabei auf der eingesetzten Methode und der Umsetzung innerhalb unserer Untersuchung. Anschließend werden die gewonnenen Ansätze zur Optimierung der Einsteiger-Freundlichkeit vorgestellt und ein allgemeines Vorgehen beschrieben, um den Transfer einer Desktop-Anwendung in eine AoW zu erleichtern.

3.1 Der Untersuchungsgegenstand

Die Methode SeeMe ermöglicht die Darstellung sozio-technischer Arbeits- und Kommunikationsprozesse. Darunter ist die Kombination von menschlichen Akteuren in sozialen Systemen mit technischen Systemen zu verstehen, also beispielsweise die Organisation eines Arbeitsprozesses mittels eines technischen Systems, an dem mehrere Personen beteiligt sind. Die Methode wurde entwickelt, um die technische Entwicklung und das soziale System integriert betrachten zu können.

Als Grundlage für die Exploration diente der auf der Modellierungsmethode SeeMe basierende Desktop Editor. Ziel war es, eine Umgebung zu schaffen, die die gemeinsame Arbeit an einem Modell entsprechend des zuvor beschriebenen Szenarios ermöglicht, wobei besonderes Augenmerk auf einer der Aufgabe angemessenen Reduktion des Funktionsumfangs lag. Zusätzlich sollte eine Optimierung hinsichtlich der Einsteiger-Freundlichkeit stattfinden, wobei die Erwartungen der Nutzer an die Interaktion mit dem Desktop Editor zusammen mit denen an das Medium Internet betrachtet werden mussten.

3.2 Methodik

Zur Exploration des Desktop Editors vor der ersten – und des Webeditors vor der zweiten – Entwicklungsphase galt es zunächst eine geeignete Methode zu finden. Dabei stand die Frage im Vordergrund, wie es möglich ist, den Funktionsumfang des Desktop Editors so zu reduzieren, dass alle für das Szenario entscheidenden Funktionalitäten erfasst werden können. Dazu ist es von entscheidender Bedeutung, die Nutzer bei der Erstellung eines Modells zu beobachten. Während innerhalb eines Interviews häufig lediglich problematische Arbeitsabläufe zur Sprache kommen und alltägliche Funktionalitäten schlicht übersehen werden, bietet die Beobachtung der Nutzer einen weitestgehend vollständigen Überblick über die verwendeten Funktionen (Holtzblatt et al. 2005). Dazu hat sich als geeignete Methode – auch aufgrund der Möglichkeit zur konsistenten Weiterverarbeitung der während der Beobachtung gewonnenen Daten – Rapid Contextual Design (Holtzblatt et al. 2005) herausgestellt. Diese erfasst Daten in Form von Affinity Notes, welche zu Gruppen zusammengefasst und innerhalb eines Diagrammes dargestellt werden. Das entstandene Affinity Diagramm diente innerhalb unserer Untersuchung als Grundlage für die Softwareentwicklung und die Betrachtung der beschriebenen Erwartungskonflikte.

Während unserer Studie wurde die Methode in zwei Phasen mit insgesamt acht Teilnehmern eingesetzt. Während der ersten Phase wurden die Nutzer bei der Verwendung des Desktop Editors beobachtet. Dabei wurde anhand des zuvor beschriebenen Szenarios versucht, den erforderlichen Funktionsumfang des Webeditors festzulegen. In der zweiten Phase wurde der festgelegte Funktionsumfang mittels einer Beobachtung bei der Verwendung eines ersten Prototyps des Webeditors evaluiert. Weiterhin wurde der Umgang der Teilnehmer mit der Problematik der verschiedenen Erwartungshaltungen beobachtet.

Die Grundvoraussetzung zur Teilnahme an der Beobachtung bestand darin, dass den Nutzern die Modellierungsmethode SeeMe zumindest lesend vertraut war, um die Ergebnisse ausschließlich auf den Aspekt der Nutzung des Editors beziehen zu können. Die Auswahl der Teilnehmer fand zusätzlich im Hinblick auf die Frequenz ihrer Nutzung des Desktop Editors statt. Dabei wurde zwischen Vielnutzern (fester Bestandteil der täglichen Arbeit), Gelegenheitsnutzern (Einsatz in verschiedenen Kontexten) und Erstnutzern (keine vorherige Verwendung) unterschieden. Diese Unterscheidung stellt sicher, dass alle Nutzertypen berücksichtigt werden. Jede Rolle wurde von zwei Teilnehmern besetzt.

3.3 Ergebnisse der Studie

Ziel der Analyse war zunächst, gruppenübergreifend allen Aussagen der Teilnehmer gerecht zu werden. Um etwaige Konflikte auflösen zu können, wurde im Hinblick auf die Einstiegschürde den Aussagen der Erstnutzer höheres Gewicht beigemessen. Anhand der gewonnenen Daten konnte ein Prototyp entwickelt werden, dessen Funktionsumfang auf das zuvor beschriebene Szenario zugeschnitten ist. Dieser Prototyp wurde anschließend speziell im Hinblick auf verschiedene Fragestellungen bezüglich des Interaktionsdesigns evaluiert. Einige besonders prägnante Fragestellungen werden im Folgenden beispielhaft beschrieben:



Abbildung 2: Darstellung der verfügbaren Funktionalitäten als Cockpit

Bei der Erstellung des Prototyps wurde der zur Verfügung stehende Funktionsumfang dem beschriebenen Szenario entsprechend angepasst. Dabei entstand das in Abbildung 2 dargestellte Cockpit. Seine Position kann von Nutzern selbst bestimmt werden, indem Sie es an der linken oberen Schaltfläche mit der linken Maustaste greifen, die Maus bewegen und die Taste an der gewünschten Position wieder loslassen. Diese Funktionsweise wurde ausgewählt, da sich die Nutzer sowohl gegen ein statisches Menü, als auch gegen ein sich am ausgewählten Element automatisch orientierendes Menü ausgesprochen haben. Während der Evaluation stellte sich das Cockpit als intuitiv bedienbar heraus. Besonders positiv bemerk-

ten vor Allem Erstnutzer, dass ihnen jederzeit ein *vollständiger Überblick über alle zur Verfügung stehenden Funktionalitäten* gewährt wird. Daraus lässt sich ableiten, dass im Hinblick auf eine niedrige Einstiegshürde die Reduktion des Funktionsumfanges sowie eine übersichtliche Darstellung der zur Verfügung stehenden Funktionen hilfreich sind.

Im ersten Prototyp des Webeditors war die Interaktion zur Verbindung von Elemente durch sogenannte Relationen (Pfeile zwischen Elementen, siehe Abbildung 1) analog zur Desktop Anwendung gestaltet. Dabei wurde mittels eines einfachen Klicks jeweils ein Punkt gesetzt, während zum Abschluss der Relation ein Doppel-Klick erforderlich war. In der Evaluation stellte sich diese Form der Erstellung von Relationen als nicht intuitiv heraus. Die Nutzer äußerten vielmehr die Erwartung, dass Relationen sich durch das Drücken der Maustaste auf dem Startpunkt und das Loslassen derselben auf dem Endpunkt erzeugen lassen. Dabei merkten die Nutzer zusätzlich an, dass sie die Interaktionsform des *Doppel-Klicks* im Kontext einer Webseite als unüblich empfanden. Somit lässt sich feststellen, dass die Nutzer den Webeditor als Webseite wahrnahmen und ihm mit den entsprechenden Erwartungen an die Funktionsweise und Interaktionsmöglichkeiten einer Webseite begegneten.

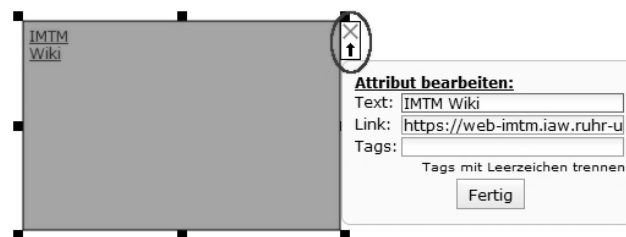


Abbildung 3: Menüanker mit Kontextmenü

Das Öffnen eines Kontextmenüs bezogen auf das im Fokus der Interaktion liegende Element erfolgte im ersten Prototyp des Webeditors über eine an dem entsprechenden Element angezeigte Interaktionsfläche (siehe Abbildung 3). Dieser Menüanker ersetzt die Interaktion des *Rechts-Klicks*, welche innerhalb der Desktop-Anwendung zum Öffnen eines Kontextmenüs verwendet wird. Während der Evaluation wurde diese Schaltfläche von allen Nutzern intuitiv verwendet. Sie fielen auf den Versuch der Rechts-Klick-Interaktion zurück, wenn der Menüanker aufgrund fehlerhaften Verhaltens des Webeditors nicht zum gewünschten Ergebnis führte. Daraus lässt sich ableiten, dass der Rechts-Klick in einer AoW nicht als Interaktionsform wahrgenommen wird, welche einen Effekt auf den dargestellten Inhalten auslöst.

3.4 Gestaltungsempfehlungen für den Transfer

Im Folgenden werden auf Basis der Ergebnisse unserer Studie Überlegungen beschrieben, die beim Transfer einer Desktop Anwendung in eine AoW berücksichtigt werden sollten. Dabei erheben wir aufgrund des nicht repräsentativen Charakters der Untersuchung keinen Anspruch auf Vollständigkeit. Vielmehr sind die Ergebnisse als Empfehlung zu verstehen, die einen ersten Vorschlag auf dem Weg zu einem allgemein gültigen Vorgehen darstellen.

Der Transfer einer Desktop Anwendung in eine gebrauchstaugliche AoW erfordert zunächst – wie in Kapitel 2 beschrieben – die Fokussierung auf einen Anwendungskontext, der an die Bedürfnisse von Erstnutzern angepasst ist. Diese müssen in die Lage versetzt werden, mit der AoW möglichst schnell möglichst viel zu erreichen. Ansonsten werden sie die AoW sehr schnell wieder verlassen und gegebenenfalls nach Alternativen suchen (Nielsen 2000). Die AoW ist also so zu gestalten, dass auch Nutzer ohne Vorkenntnisse verwertbare Ergebnisse erzielen können, die den Mehrwert der Anwendung für ihre Arbeit erkennen lassen. Dazu ist eine Reduktion des Funktionsumfangs auf die Kernfunktionalitäten der Desktop Anwendung erforderlich. Dies bedingt die Definition eines dem späteren Arbeitskontext angemessenen Szenarios, welches als Grundlage für eine entsprechende Anpassung des Funktionsumfangs dient. In unserem Fall wurde dazu der Funktionsumfang des Desktop Editors im Hinblick auf die Möglichkeit zum schnellen grafischen skizzieren von Modellen angepasst. Dazu entstand das in Abbildung 2 rechts dargestellte Menü. Dieses ist so gestaltet, dass es dem Nutzer jederzeit einen Überblick über alle verfügbaren Funktionalitäten bietet.

Anschließend ist es notwendig, die identifizierten Kernfunktionalitäten so zu gestalten, dass der Nutzer in der Lage ist, diese intuitiv zu verwenden. Die Oberfläche ist also so zu gestalten, dass die zur Verfügung stehenden Funktionalitäten leicht erreichbar und übersichtlich dargestellt sind. Weiterhin sind beim Design der Interaktionen die Nutzererwartungen an das Medium Internet zu berücksichtigen, was in unserer Untersuchung einen gänzlichen Verzicht auf die Interaktionsform des Recht- und Doppel-Klicks zur Folge hatte. Die entsprechenden Funktionalitäten wurden durch Modifikation der Interaktion (Erzeugung einer Relation) und die Einführung zusätzlicher Schaltflächen (Menüanker) abgedeckt.

4 Vergleich mit vorhandenen Ansätzen

Die uns bekannte einschlägige Literatur lässt Ansätze zum Transfer einer Desktop Anwendung in eine gebrauchstaugliche AoW gänzlich vermissen. So beschäftigen sich zwar verschiedene Ansätze mit den Auswirkungen des Phänomens Web 2.0 auf den Aspekt Usability, betrachten dabei jedoch weniger die speziellen Erfordernisse einer AoW. Vielmehr stehen die Auswirkungen der Möglichkeit zur Gestaltung der Oberfläche durch die Nutzer im Vordergrund. Diese erhöhen häufig lediglich die Komplexität der Interaktion ohne tatsächlich nützlich zu sein (Nielsen 2007). Weiterhin beeinträchtigen allzu weitreichende Gestaltungsmöglichkeiten der Oberfläche die Konsistenz der Darstellung (Dix & Cowen 2007). Daraus resultiert, dass dem Nutzer die zur Verfügung stehenden Interaktionsmöglichkeiten nicht mehr bewusst sind (Pilgrim 2008). Dadurch entsteht ein Konflikt zwischen den Erwartungen der Nutzer und der tatsächlichen Darstellung, was in einer kognitiven Überlastung mündet.

Die Anforderungen beim Transfer einer Desktop Anwendung in eine AoW werden jedoch hinsichtlich der benötigten Anpassung des Funktionsumfangs sowie der Lösung des Erwartungskonfliktes nicht betrachtet. Diese Aspekte konnten in unserer Analyse erforscht und es konnten erste Ergebnisse zur Lösung des Problem vorgestellt werden.

Im Vergleich der hier vorgestellten Arbeit zu anderen AoW wie beispielsweise Google Docs lässt sich zunächst feststellen, dass auch bei diesen eine Reduktion auf die Kernfunktionalitäten der jeweiligen Anwendung stattgefunden hat. Dabei stand offensichtlich – wie auch in unserer Untersuchung – eine Fokussierung auf unmittelbare Nutzbarkeit im Vordergrund.

Zusätzlich fällt auf, dass hinsichtlich der Erwartungen an eine Desktop Anwendung im Vergleich zu einer AoW kein konsistentes Interaktionsdesign erkennbar ist. Bei Google Docs kann innerhalb Dokumentenübersicht mittels eines einfachen Klicks das ausgewählte Dokument geöffnet werden. Eine Interaktion also, die im Web üblich und in einer Desktop Anwendung durch einen Doppel-Klick ausgelöst würde. Befindet man sich nun innerhalb des gewählten Dokumentes, so besteht die einzige Möglichkeit, ein Kontextmenü zu einem entsprechenden Element zu öffnen, darin, sich eines Rechts-Klicks zu bedienen. Eine Interaktion, die in einer Desktop Anwendung absolut gängig und im Web total unüblich ist. Komplette wird die Konfusion an der Stelle, wo das Öffnen eines Kontextmenüs innerhalb der Tabellenkalkulation anstandslos funktioniert, während in der Textverarbeitung lediglich das Kontextmenü des Browsers angezeigt wird.

5 Zusammenfassung und Ausblick

Im Zuge unserer Untersuchung konnte der Funktionsumfang des SeeMe Desktop Editors im Hinblick auf das in Kapitel 3 beschriebene Szenario angepasst werden. Weiterhin wurde beim Design der Oberfläche und Interaktionen der Erwartungskonflikt der Nutzer zwischen einer Desktop Anwendung und einer AoW berücksichtigt. Schließlich konnten Möglichkeiten beschrieben werden, die beim Transfer einer beliebigen Desktop Anwendung in eine AoW eine erwartungskonforme Umsetzung unterstützen. Aufgrund des explorativen Charakters unserer Untersuchung kann jedoch kein Anspruch auf Vollständigkeit erhoben werden.

Zukünftig wäre es interessant, den Einfluss von Vorerfahrungen der Nutzer auf seine Erwartungshaltung im Bezug auf die Interaktion mit einer AoW zu erforschen. Im Fokus der Betrachtung kann dabei speziell die Frage stehen, ob Vorerfahrungen mit anderen Webseiten oder Desktop Anwendungen – wie beispielsweise Textverarbeitungs-Software – die Erwartungen der Nutzer beeinflussen. Zusätzlich stellt sich die Frage, ob Nutzer beim Übergang zu Funktionalitäten des Web 2.0 bereit sind, ihre Erwartungen entsprechend anzupassen. Auch die Frage in wieweit die Integration einer AoW in eine umgebende Webseite Einfluss auf die Erwartungen der Nutzer ausübt, könnte sich als spannend erweisen, da die aktuell verbreiteten AoW sich als einziges Element innerhalb des Browser-Fensters präsentieren. So könnte beispielsweise die Integration des von uns entwickelten Webeditors in eine Wiki die Nutzerwahrnehmung und seine Erwartungen an die Interaktion massiv beeinflussen.

Literaturverzeichnis

- Cockton, G., Lavery, D. & Woolrych, A. (2003). Inspection-based evaluations. In Jacko, J. A. & Sears, A. (Hrsg.): *The human-computer interaction handbook: Fundamentals, evolving technologies, and emerging applications*. London: Lawrence Erlbaum Associates, S. 1118-1138.
- Dix, A. & Cowen, L. (2007). HCI 2.0? Usability meets Web 2.0. Beitrag zum Panel BCS HCI 2007

- DIN EN ISO 9241-11 (1998). *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze*. Beuth, Berlin.
- DIN EN ISO 9241-110 (2006). *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 110: Grundsätze der Dialoggestaltung*. Beuth, Berlin.
- Galitz, W. O. (2002). *The Essential Guide to User Interface Design. Second Edition*. New York: Wiley Computer Publishing.
- Herrmann, T. (2009). Systems Design with the Socio-Technical Walkthrough. In Brian Whitworth & Aldo de Moor (Hrsg.): *Handbook of Research on Socio-Technical Design and Social Networking Systems*, Hershey: Idea Group Publishing, S. 336-351.
- Holtzblatt, K., Wendell, J. B. & Wood, S. (2005). *Rapid Contextual Design. A How-To Guide to Key Techniques for User-Centered Design*. San Francisco: Morgan Kaufmann Publishers.
- McAfee, A. P. (2006). Enterprise 2.0: The Dawn of Emergent Collaboration. *MIT Sloan Management Review Spring 2006 Volume 47 No. 3*, S. 21-28.
- Nielsen, J. (2000). *Designing Web Usability*. Indianapolis: New Riders Publishing
- Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In Adelson, B., Dumais, S. T. & Olson, J. S. (Eds.): *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York: ACM Press, S. 152-158
- Nielsen, J. (2007). Web 2.0 Can Be Dangerous. In *Jakob Nielsen's Alertbox* abgerufen am 03.03.2009. <http://www.useit.com/alertbox/web-2.html>
- O'Reilly, T. (2007). What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software. *International Journal of Digital Economics 2007 65*, S. 17-37
- Pilgrim, C. J. (2008). Improving the usability of web 2.0 applications. *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*. New York: ACM Press, S. 239-240

Kontaktinformationen

Alexander Nolte

Informations- und Technikmanagement
Ruhr-Universität Bochum
D- 44780 Bochum
Universitätsstraße 150
Tel.: +49 (0)234 / 32-22045
E-Mail: nolte@iaw.rub.de

Michael Prilla

Informations- und Technikmanagement
Ruhr-Universität Bochum
D- 44780 Bochum
Universitätsstraße 150
Tel.: +49 (0)234 / 32-27735
E-Mail: prilla@iaw.rub.de