

Hans-Christoph Hobohm (Hrsg.)

Informationswissenschaft zwischen virtueller Infrastruktur und materiellen Lebenswelten

**Information Science between
Virtual Infrastructure and Material Lifeworlds**

Unter Mitarbeit von Judith Pfeffing

Proceedings des 13. Internationalen Symposiums
für Informationswissenschaft (ISI 2013)

Potsdam, 19. bis 22. März 2013

vwh

Verlag Werner Hülsbusch
Fachverlag für Medientechnik und -wirtschaft

Usability-Probleme bei Desktop-Suchmaschinen

Manuel Burghardt¹, Jürgen Reischer², Christian Wolff¹

Universität Regensburg

¹ Lehrstuhl für Medieninformatik

² Lehrstuhl für Informationswissenschaft

{manuel.burghardt|juergen.reischer|christian.wolff}@ur.de

Zusammenfassung

Desktop-Suchmaschinen (DSM) unterscheiden sich in wesentlichen Punkten von Web-Suchmaschinen (WSM), weshalb ihr Interaktionsdesign anderen Anforderungen genüge leisten muss. Der Artikel stellt eine expertenbasierte Usability-Studie vor, bei der insgesamt acht DSM mithilfe eines ‚Heuristic Walkthrough‘ untersucht wurden. Die dabei identifizierten Usability-Probleme werden beschrieben und mögliche Lösungsansätze diskutiert.

Abstract

Desktop search engines (DSE) are an important tool for ‘personal information management’ (PIM) as they offer full text search capabilities comparable to search engines for the World Wide Web. In this paper we discuss problems of DSE interface design and present results from an investigation of the usability qualities of various DSE using the heuristic walkthrough method. We discuss typical design problems as well as possible solutions.

In: H.-C. Hobohm (Hrsg.). Informationswissenschaft zwischen virtueller Infrastruktur und materiellen Lebenswelten. Tagungsband des 13. Internationalen Symposiums für Informationswissenschaft (ISI 2013), Potsdam, 19.–22. März 2013. Glückstadt: Verlag Werner Hülsbusch, 298–309.

1 Einleitung

Obwohl das Web und die Web-Suche zu den größten Erfolgsgeschichten der Informationstechnologie überhaupt zählen, und Suchmaschinen wie Google tagtäglich zur Informationssuche genutzt werden, scheint die Nutzung von Tools zum Durchsuchen des eigenen Rechners weitaus weniger verbreitet und etabliert zu sein (Liao 2007). Dabei sollen sogenannte Desktop-Suchmaschinen (DSM, vgl. Cole 2005) in Analogie und Erweiterung zu Web-Suchmaschinen (WSM) die persönlichen Datenbestände auf privaten Rechnern durchsuchbar machen, und dem Nutzer bei der Organisation und Wiederfindung persönlicher Informationsobjekte helfen. Weder die mittlerweile auf allen gängigen Betriebssystemplattformen verfügbare integrierte Volltextsuch- bzw. Suchmaschinenfunktionalität noch als dedizierte Zusatzsoftware zu installierende Lösungen erreichen in der Praxis den gewünschten Nutzungsgrad, wobei Literatur zu diesem Thema sich mehr oder weniger auf populärwissenschaftliche Testberichte beschränkt (vgl. Posey 2012). Dabei ist die Fokussierung auf Desktop-Rechner bzw. die Desktop-Metapher an sich aufgrund des Einzugs mobiler Endgeräte mit neuen Verfahren der Interaktion bereits überholt. Der Begriff ‚desktop search‘ ist mittlerweile ambig, da einerseits auf die Suche in lokalen Datenbeständen (also auf PIM, ‚personal information management‘) Bezug genommen wird, andererseits unter ‚desktop search‘ diejenige Nutzung von Suchmaschinen im Web verstanden wird, die von einem Desktop-Rechner aus gestartet wird – im Unterschied zur ‚mobile search‘. Denkt man noch die zunehmende Verlagerung der eigenen Daten in die Cloud hinzu, ergeben sich für Suchprobleme mehrere Dimensionen: eigene oder fremde Daten, Lokalisation der Daten auf eigenem Rechner oder im Netz/in der Cloud, Durchführung der Suche von einem Desktop oder einem mobilen Gerät aus. Da viele Nutzer mehrere stationäre und mobile Geräte besitzen und parallel betreiben, werden persönliche Daten zunehmend fragmentiert und schwerer auffindbar. Systeme zum ‚personal information management‘ (PIM) bzw. ‚personal information access‘ (Elsweiler et al. 2010) sollen den Nutzer dabei unterstützen, seine eigenen Informationsobjekte zu verwalten und verfügbar zu machen – unabhängig von Speicherort und Art des benutzten Endgerätes.

1.1 Unterschiede zwischen Web- und Desktop-Suche

Oftmals findet man Information im Web besser und schneller als persönliche Daten und Dokumente auf eigenen Rechnern (Dumais et al. 2003). Das Konzept, die Suche in privaten Datenbeständen auf dem eigenen Desktop analog zur Suche in den öffentlichen Datenbeständen des Webs zu gestalten, scheint daher naheliegend, ist jedoch in seiner jetzigen Form wenig erfolgreich. Die Gründe hierfür mögen unter anderem darin liegen, dass es nicht nur unterschiedliche Charakteristika innerhalb des Web-Suchparadigmas gibt (Wolf-ram 2008), sondern sich Web- und Desktop-Suche deutlich voneinander unterscheiden (vgl. Dumais et al. 2003; Cole 2005; Cutrell/Dumais/Teeven 2006). Die wesentlichen Unterschiede verdeutlicht Tabelle 1.

Tab. 1: Grundlegende Unterschiede zwischen Web- und Desktop-Suche

Web-Suche	Desktop-Suche
Information überwiegend im Format vorstrukturierter und standardisierter HTML-Dokumente (tendenziell eher homogene Strukturen)	Information in nicht vorstrukturierten Dokumenten unterschiedlicher, teilweise proprietärer Formate (tendenziell eher heterogene Strukturen)
Dokumente als Hypertext verlinkt mit formalem und inhaltlichem Zusammenhang	einzelne/isolierte Dokumente oftmals ohne inhaltlichen Zusammenhang
Suche nach einer bestimmten Information, nicht nach einem Dokument (Ort der Information irrelevant)	Suche nach einem Dokument oder einer Information in einem Dokument (Ort der Information relevant)
oftmals Suche nach unbekannter, neuer Information (häufig noch nicht zuvor gesehene Inhalte)	meist Suche nach bekannter Information oder bekanntem Dokument („stuff I’ve seen“)
praktisch unbeschränkte Rechenleistung der Serverfarmen des Suchmaschinenbetreibers (globaler Index)	Suche mit beschränkter Rechenleistung des eigenen Rechners (lokaler Index)
hohe Rate an Informationsupdates (sekündlich, minütlich)	niedrige Rate an Informationsupdates (stündlich, täglich)
Suche nach öffentlicher/veröffentlichter Information	Suche nach persönlicher Information, die nicht öffentlich werden soll (privacy, security)

Durch Cloud-Technologien, bei der persönliche Informationen zentral auf Servern im Netz bereitgestellt werden, fließen Web- und Desktop-Suche (bzw. „personal information access“) bis zu einem gewissen Grad zusammen:

Der Nutzer besitzt eine Nische im Web, in der seine persönliche Information gelagert und nur ihm über unterschiedliche Endgeräte zur Verfügung gestellt wird. Die Auslagerung von Information ins private Netz stellt dabei eine weitere Herausforderung für PIM-Systeme dar, als diese zusätzlich mit lokal und global verstreuten Informationsobjekten umgehen können müssen.

1.2 Nutzerverhalten bei der Desktop-Suche

Bergman et al. (2008) untersuchen Strategien und Präferenzen im Nutzerverhalten bei der Suche nach persönlicher Information auf privat genutzten, mobilen oder stationären Computersystemen im Rahmen des PIM. Sie unterscheiden zwischen ‚hierarchischer Navigation‘, überwiegend realisiert durch baumartig organisierte Ordner- und Dateistrukturen, und ‚direkter‘ oder ‚gezielter Suche‘ mittels Suchanfrage, durch die ein Attribut des gesuchten Informationsobjekts adressiert wird (z. B. enthaltene Wörter oder Dateiname/ Datum einer Datei). Dabei stellt sich heraus, dass die überwiegende Mehrheit der Nutzer trotz Fortschritten in der Technologie der PIM-Systeme nach wie vor die hierarchische Navigation der Suche vorzieht; die Suche wird meist nur als letzter Ausweg („last resort“: Bergman et al. 2008: 20) genutzt, wenn der Ort des Informationsobjekts in der hierarchischen Struktur nicht mehr präsent ist. Die Gründe hierfür liegen vor allem in der Art und Weise der Informationsverarbeitung der Nutzer: Die Navigation in selbsterstellten, konsistenten und persistenten Ordnerhierarchien folgt nicht nur der eigenen Denkstrategie über die Organisation von Informationseinheiten, sie wird durch die Vertrautheit mit der Struktur auch kognitiv automatisiert. Bei der Suche hingegen müssen explizit Eigenschaften des gesuchten Objekts aus dem Gedächtnis rekapituliert werden („recall“), bei der Navigation bilden bereits die Ordnerstruktur und die Bezeichnungen der Ordner die im Gedächtnis des Nutzers vorhandenen Strukturen nach, wobei er die gewünschte Information nur mehr erkennen und auswählen muss („recognition“).

1.3 Zielsetzung und Aufbau der Studie

Worin gründet die mangelnde Akzeptanz von DSM bzw. PIM-Systemen? Die Unterschiede zwischen Desktop- und Web-Suche liefern zwar Anhaltspunkte für grundlegende Unterschiede im Nutzungskontext von Suchmaschinen, sie geben jedoch noch keine hinreichenden Gründe für den geringen

Erfolg von PIM-Systemen: Diese liegen zum einen im Nutzer und seinem Suchverhalten (siehe oben, Abschnitt 1.2). Daneben gehen wir davon aus, dass auch die konkrete Systemgestaltung der DSM und ihre Gebrauchstauglichkeit die Akzeptanz und Nutzungsintensität beeinflussen: Die Informationssuche auf einem lokalen, persönlichen und vertrauten System unterscheidet sich grundlegend von der Suche im ‚unbekannten‘ World Wide Web. Ein System, das den Benutzer dabei unterstützt, seinen privaten Rechner zu durchsuchen, muss deshalb auch andere Anforderungen an das Interfacedesign erfüllen, als dies bei WSM der Fall ist. In einer Usability-Studie untersuchen wir, ob es auffällige oder sogar wiederkehrende Usability-Probleme bei unterschiedlichen verfügbaren DSM gibt, deren Behebung die allgemeine Benutzbarkeit und Benutzerfreundlichkeit in Zukunft steigern kann.¹ Nachfolgend geben wir zunächst einen knappen Überblick zu geeigneten Verfahren des Usability Testing (Kap. 2), erläutern unser Versuchsdesign (Kap. 3) und stellen die wichtigsten qualitativen Ergebnisse vor (Kap. 4). Ein kurzer Ausblick zeigt weiteren Forschungsbedarf auf.

2 Usability und Usability-Engineering

‚Usability‘, im Deutschen häufig mit Benutzerfreundlichkeit übersetzt, gibt an, wie gut bzw. intuitiv ein System benutzbar ist. Usability lässt sich mit Methoden des Usability Engineering systematisch umsetzen und testen (vgl. ISO 9241-210: Human-centered design of software). Das Usability Testing (d. h. die Evaluation interaktiver Systeme) ermöglicht es, Probleme frühzeitig zu identifizieren und zu beheben. Das Spektrum hierfür verfügbarer Evaluationsmethoden kann grob in zwei Hauptkategorien unterteilt werden (Rosson/Carroll 2002): Endnutzerorientierte empirische Methoden sammeln Informationen über ein System, indem reale Nutzer beobachtet und befragt werden. Analytische Methoden hingegen beschreiben die Evaluation eines Systems durch Usability-Experten, die sich durch Kenntnis der Nutzeranforderungen der Anwendungsdomäne so gut wie möglich in Endnutzer hineinversetzen. Eine Variante der analytischen Methoden sind die Inspektionsmethoden, zu

¹ Die Bewertung der Qualität der Suchergebnisse war nicht Ziel der Untersuchung.

denen unter anderem der ‚Cognitive Walkthrough‘ (CW) sowie die ‚Heuristische Evaluation‘ (HE) gehören.

Bei einem CW versucht der Evaluator, ein Höchstmaß an Empathie mit der tatsächlichen Nutzergruppe herzustellen, indem er sich immer wieder Kontrollfragen stellt, z. B. „Weiß der Benutzer, was als nächstes zu tun ist, um die gestellte Aufgabe zu erreichen, d. h. wird der Benutzer vom System geführt?“ (Wharton et al. 1994). Während der Durchführung des CW erfolgt eine grundlegende Systemexploration durch das Erfüllen vordefinierter Aufgaben, welche von den Evaluatoren eigenständig dokumentiert wird.

Die HE beschreibt eine relativ unstrukturierte Expertenevaluation eines Systems, bei der eine Sammlung von Usability-Prinzipien, die Heuristiken, als Evaluationsgrundlage dient. Zu den am weitesten verbreiteten Heuristiken zählen Ben Shneidermans „Eight Golden Rules of Interface Design“ (Shneiderman/Pleasant 2009) und Jakob Nielsens „Ten Usability Heuristics“ (Nielsen 1994). Den Heuristiken kommt eine Art Checklisten-Funktion zu, die den Evaluator für konkrete Usability-Probleme sensibilisieren soll, und ihm hilft, die identifizierten Probleme grundlegend zu kategorisieren.

Mit dem ‚Heuristischen Walkthrough‘ (HW) (Sears 1997) liegt schließlich eine Kombination aus CW und HE vor, d. h. zunächst exploriert der Evaluator das System mithilfe der Methode des CW und dokumentiert dabei die einzelnen Schritte. Im Anschluss erfolgt die Evaluation des zuvor explorierten Systems mithilfe einer Sammlung von Heuristiken. Zuletzt werden die Evaluationsergebnisse ausgewertet und ggf. zusammengefasst, um eine Liste von Usability-Problemen zu erhalten.

3 Evaluationsdesign

Um die Usability von Systemen zur Desktop-Suche zu evaluieren, wurde der zweistufige Ansatz des HW verfolgt. Da es für den Bereich der DSM kein etabliertes Heuristiken-Set gibt, wurden die allgemein verwendbaren Usability-Heuristiken nach Nielsen (1994) verwendet (Übersetzung nach Meiert 2006):

- Sichtbarkeit des Systemstatus
- Übereinstimmung zwischen System und Wirklichkeit
- Nutzerkontrolle und -freiheit

- Konsistenz und Standards
- Fehlervorbeugung
- Erkennen anstatt Erinnern
- Flexibilität und Effizienz
- ästhetisches und minimalistisches Design
- Benutzern sollte geholfen werden, Fehler zu erkennen, zu diagnostizieren, und sich von diesen wieder zu „erholen“
- Hilfe und Dokumentation

Insgesamt wurden acht DSM ausgewählt, die entweder bereits in gängige Betriebssysteme integriert sind oder als add-on-Tools wenigstens in einer Basisvariante kostenfrei installiert werden können (vgl. Tab. 2). Alle Systeme wurden von jeweils 1–2 Evaluatoren untersucht. Die Evaluation wurde von Studierenden im Rahmen einer Lehrveranstaltung des Fachs Informationswissenschaft durchgeführt.

Tab. 2: Getestete Desktop-Suchmaschinen

Name der DSM	URL
X1	http://www.x1.com/
Filehand Search	http://www.filehand.com/
Copernic Desktop Search	http://www.copernic.com/en/products/desktop-search/index.html
Windows Search 4.0	http://www.microsoft.com/de-de/download/details.aspx?id=23 (standardmäßig in das Betriebssystem integriert seit Windows 7)
Archivarius 3000	http://www.likasoft.com/de/document-search/index.shtml
Everything	http://www.voidtools.com/
xFriend	http://www.xfriend.de/
Apple Spotlight	standardmäßig in das Betriebssystem integriert seit Mac OS X 10.4

Die mögliche Verzerrung der Ergebnisse durch die inkonsistente Anzahl an Evaluatoren ist für das Erkenntnisinteresse dieser Studie unerheblich, da es nicht darum geht, das beste, also benutzerfreundlichste, Tool zu finden. Vielmehr sollen möglichst viele Usability-Mängel identifiziert werden, die typisch für den Bereich der DSM sind. Die Evaluation der acht DSM wurde jeweils auf unterschiedlichen Rechnern durchgeführt, die aber alle die empfohlenen Systemvoraussetzungen der jeweiligen Suchmaschine erfüllten. Jeder der Evaluatoren erhielt dieselbe Aufgabenstellung und Einführung in das

Evaluationsverfahren (zusätzlich als Evaluationsleitfaden dokumentiert und jederzeit während der Evaluation nachlesbar). Zur Aufgabenstellung gehörten außerdem einige Dateien, welche die Evaluatoren auf ihrem jeweiligen Testrechner ablegten und von der Suchmaschine indexieren ließen. Bei der Konzeption der Aufgaben sowie der entsprechenden Testdokumente wurde im Vorfeld sichergestellt, dass die Evaluatoren keine Dokumente auf ihrem System haben, die mit den Testanfragen kollidieren,² und somit jeder Evaluator (je nach Qualität der jeweiligen Suchmaschine) dieselbe Ergebnismenge erhielt. Während der Systemexploration, die durch das Abarbeiten der Aufgaben gewährleistet wurde, dokumentierten die Testpersonen ihr Vorgehen, um daraus im letzten Schritt Usability-Probleme abzuleiten.

4 Diskussion der Ergebnisse

Der folgende Abschnitt diskutiert die häufigsten Usability-Probleme sowie mögliche Lösungsansätze und Implikationen für künftige DSM.

4.1 Quantitative Ergebnisse

Insgesamt wurden für die acht DSM von den Evaluatoren 70 Usability-Probleme identifiziert. Da manche Usability-Probleme gegen mehr als eine Heuristik verstoßen, konnten insgesamt 92 Verstöße gegen Nielsens Usability-Heuristiken dokumentiert werden. Dabei fällt auf, dass jede der zehn Heuristiken mehrfach verletzt wurde, d. h. es konnte ein grundlegender Optimierungsbedarf von DSM für die ganze Bandbreite von Usability-Heuristiken aufgezeigt werden. Gleichzeitig gibt es weder nach oben noch nach unten Ausreißer, d. h. keine der Heuristiken beschreibt einen Usability-Bereich, der besonders kritisch oder unkritisch für DSM wäre. Usability-Evaluationen, die mithilfe nur einer Inspektionsmethode durch wenige Experten durchgeführt wurden, liefern keine statistisch belastbaren, quantitativen Daten, sondern qualitative Ergebnisse, die im Folgenden diskutiert werden.

² 2 Aufgaben und Testdokumente beschäftigten sich mit dem Thema „Fliegenfischen“ – alle Evaluatoren bestätigten, außer den Testdokumenten keine anderen Dokumente zu diesem Thema auf ihrem System gespeichert zu haben.

4.2 Qualitative Ergebnisse

Die Auswertung³ der Evaluationsergebnisse zeigt, dass viele der insgesamt 70 identifizierten Usability-Probleme inhaltlich übereinstimmen, d. h. es gibt einige Usability-Probleme, die von unterschiedlichen Evaluatoren für unterschiedliche DSM identifiziert wurden. Offensichtlich gibt es Probleme, die in verschiedenen DSM immer wiederkehren, und die deshalb spezifisch für diesen Untersuchungsbereich sind. Gleichzeitig findet sich eine Reihe von Usability-Problemen, die nicht spezifisch für den Bereich der DSM sind, sondern vielmehr für allgemeine Usability-Aspekte wie schlechtes Icon-Design oder unangemessenes Layout der Menüstruktur stehen. Im Folgenden werden Designempfehlungen zur Lösung wiederkehrender, DSM-spezifischer Usability-Probleme beschrieben:⁴

- (a) *Reichweite der Indexierung*: Die Benutzer wollen wissen, welche Bereiche des Systems indexiert werden, und sie wollen die Indexierung bei Bedarf konfigurieren können, um z. B. persönliche Ordner von der Indexierung auszunehmen. Die DSM sollte deutlich machen, welche Bereiche des Systems anfänglich indexiert werden, und dem Benutzer erlauben, bei Bedarf die Reichweite der Indexierung individuell zu konfigurieren.
- (b) *Zeitpunkt der Indexierung*: Die Benutzer wollen wissen, wann das System jeweils indexiert, und wie sich das auf die Performanz der Suche, aber auch auf die Qualität der Ergebnisse auswirkt. Die DSM sollte anzeigen, wann sie den Index aktualisiert, und ob der Index gerade auf dem aktuellen Stand ist.
- (c) *Alternativvorschläge*: Falls die DSM bei einer Suchanfrage keine Treffer liefert, sollte das System Feedback statt einer leeren Ergebnisseite geben. Das Feedback kann Hinweise auf Schreibvarianten oder verwandte Suchterme enthalten oder Tipps für die Umformulierung von Anfragen geben.
- (d) *Duplikate*: Falls ein Dokument mehrfach auf dem System gespeichert ist, sollten die Duplikate in der Ergebnisliste entsprechend visualisiert werden.

³ Die Auswertung der inhaltlichen Zusammenfassung erfolgte durch die Autoren.

⁴ Analog dazu sei auf die Checkliste von Quirnbach (2012: 159 ff.) verwiesen, welche eine umfangreiche Sammlung generischer Gestaltungshinweise für ein benutzerfreundliches Suchmaschinen-Design enthält, allerdings nicht explizit für den spezifischen Kontext der Desktop-Suche konzipiert wurde.

- (e) *Ergebnisdarstellung*: Viele Nutzer fühlen sich unsicher, wenn sie sowohl vor als auch nach der Suche nicht wissen, wo ein bestimmtes Dokument auf dem System tatsächlich gespeichert ist. Der Benutzer will Dokumente für seine Suchanfrage nicht nur direkt öffnen können, sondern auch wissen, wo die Dokumente auf seinem System abgelegt sind, z. B. durch Angabe des vollständigen Pfads im Dateisystem.
- (f) *Trefferliste*: Die Dokumente der Ergebnisliste sollten mit relevanten Metadaten versehen sein, um dem Benutzer die Interpretation der Treffer zu erleichtern (z. B. Unterscheidung zwischen dem Dokumentformat PDF und dem Grafikformat PNG). Der Benutzer sollte außerdem mit einem Klick auf den jeweiligen Treffer zum Originaldokument springen können.
- (g) *Speichern von Anfragen*: Die Benutzer von DSM wollen bewusst immer wieder gleiche Suchanfragen stellen, um an bestimmte Dokumente zu gelangen (ähnlich wie Verknüpfung oder Alias erstellen), möglicherweise weil sie sich die erfolgreiche Suchanfrage besser merken können als den tatsächlichen Ablageort. In diesem Fall liegt – anders als oben in Kap. 1 beschrieben – eine Präferenz für die Suche gegenüber der Navigation im Dateisystem vor. Anders als im Web kann sich der Benutzer sicher sein, dass gleiche Suchanfragen auch gleiche Ergebnisse bringen, sofern er nicht selbst die Dokumentmenge verändert hat. Deshalb sollten DSM vor allem komplexe Suchanfragen speichern und verwalten können.
- (h) *Suchoperatoren*: Bei der Benutzung von WSM haben sich mittlerweile einige Konventionen bezüglich der Suchoperatoren herausgebildet (Jansen/Spink 2006). So bedeutet bei Google und anderen bekannten WSM bspw. die Aneinanderreihung mehrerer Terme, dass diese mit dem AND-Operator verknüpft werden sollen, wenn auch nicht notwendigerweise in dieser Reihenfolge, was erst durch einen Phrasen-Operator (typischerweise doppelte Anführungszeichen realisiert) erzwungen wird. Solche Konventionen aus dem Web-Bereich sollten auch die DSM beachten.
- (i) *Search as you type*: Eine weitere Konvention, welche die Benutzer von WSM kennen, ist die ‚search as you type‘-Funktion, bei der ohne explizites Absenden der Abfrage bei der zeichenweisen Eingabe eine jeweils veränderte Ergebnisliste angezeigt wird und so dem Benutzer ggf. das weitere Vervollständigen der Suchanfrage erspart bleibt. Falls search as you type nicht vorhanden ist, sollte zumindest die ursprüngliche Suchanfrage auch auf der Ergebnisseite noch sichtbar sein, da die Benutzer bei einer unbefriedigenden Trefferliste häufig an der richtigen Formulierung

bzw. der Richtigkeit der ursprünglich beabsichtigten Suchanfrage zweifeln.

- (j) *Suchstatus*: Da im Gegensatz zu WSM, wo leistungsfähige Server die Anfrage der Nutzer verarbeiten, die Geschwindigkeit von DSM stark von der Hardware des jeweiligen Benutzers abhängig ist, dauern Suchanfragen bei Desktop-Systemen oftmals länger, als man dies von der Web-Suche gewohnt ist. Der Suchfortschritt sollte für den Benutzer jederzeit sichtbar sein. Ebenso sollte der Suchprozess vom Benutzer manuell abgebrochen werden können.
- (k) *Web-Suche*: Falls eine DSM gleichzeitig die Möglichkeit bietet, auch das Web zu durchsuchen, sollte für den Benutzer eindeutig ersichtlich sein, welche Dokumente in der Trefferliste sich wo befinden. In der Evaluation zeigte sich, dass viele Benutzer kognitiv überlastet sind, wenn Sie eine kombinierte Ergebnisliste der persönlichen Desktop-Treffer und gleichzeitig relevanter Web-Treffer interpretieren müssen.

5 Ausblick

Auf Basis der hier vorgestellten Usability-Studie soll in einem nächsten Schritt eine Usability-Heuristik erstellt werden, die einerseits als Evaluationsinstrument und andererseits als Design-Guideline für die Entwicklung von DSM dient. Tooldesigner erhalten damit eine Handreichung, die sie bereits früh im Entwicklungsprozess für die wichtigsten Usability-Probleme sensibilisiert. Gleichzeitig kann die speziell für den Bereich der Desktop-Suche erstellte Heuristik für die Evaluation weiterer DSM eingesetzt werden, und so Usability-Probleme bestehender Systeme aufzeigen.

Literaturverzeichnis

- Bergman, O.; Beyth-Marom, R.; Nachmias, R.; Gradovitch, N.; Whittaker, S. (2008). Improved search engines and navigation preference in personal information management. In: ACM Transactions on Information Systems (TOIS) 26 (1), 1–24.
- Cole, B. (2005). Search engines tackle the desktop. In: Computer 38 (3), 14–17.

- Cutrell, E.; Dumais, S. T.; Teevan, J. (2006). Searching to Eliminate Personal Information Management. In: *Communications of the ACM* 49 (1), 58–64.
- Dumais, S.; Cutrell, E.; Cadiz, J. J.; Jancke, G.; Sarin, R.; Robbins, D. C. (2003). Stuff I've seen: a system for personal information retrieval and re-use. In: *Proc. SIGIR '03. 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 72–79.
- Elsweiler, D.; Jones, G. J. F.; Kelly, L.; Teevan, J. (2010). Workshop on desktop search. In: *Proc. SIGIR '10. 33rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Genf, Juli 2010. New York: ACM, online: <http://www.cdvpc.dcu.ie/DS2010/> <19.02.2013>.
- ISO 9241-11. (1999). Ergonomic requirements for office work with visual display terminals – Part 11: Guidance on usability.
- Jansen, B. J.; Spink, A. (2006). How are we searching the World Wide Web? A comparison of nine search engine transaction logs. In: *Information Processing & Management* 42 (2), 248–263.
- Liao, F. (2007). Are you using Desktop Searching? <http://www.pimagenda.com/2007/11/21/are-you-using-desktop-searching/> <7.01.2013>.
- Meiert J. (2006). Usability-Heuristiken. <http://meiert.com/de/publications/articles/20051218/> <7.1.2013>.
- Nielsen, J. (1994). Heuristic evaluation. In: J. Nielsen; R. Mack (Eds.), *Usability Inspection Methods*. New York: Wiley & Sons, 25–62.
- Posey, B. (2012). Five fast Windows desktop search utilities. TechRepublic Website, August 2012. <http://www.techrepublic.com/blog/five-apps/five-fast-windows-desktop-search-utilities/1512> <7.1.2013>.
- Quirnbach, S. (2012). *Suchmaschinen*. Heidelberg: Springer.
- Rosson, M. B.; Carroll, J. M. (2002). *Usability Engineering. Scenario-based development of human computer interaction*. San Francisco u. a.: Morgan Kaufman.
- Sears, A. (1997). Heuristic walkthroughs: finding the problems without the noise. In: *International Journal of Human-Computer Interaction* 9 (3), 213–234.
- Shneiderman, B.; Plaisant, C. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Amsterdam: Addison Wesley.
- Wharton, C.; Riemann, J.; Lewis, C.; Polson, P. (1994). The cognitive walkthrough method: a practitioner's guide. In: Nielsen, N.; Mack R. (Eds.). *Usability Inspection Methods*. New York: Wiley & Sons, 105–140.
- Wolfram, D. (2008). Search characteristics in different types of Web-based IR environments: Are they the same? In: *Information Processing & Management* 44 (3), 1279–1292.