

HyperScout: Darstellung erweiterter Typinformationen im World Wide Web – Konzepte und Auswirkungen

Harald Weinreich, Hartmut Obendorf, Winfried Lamersdorf
Universität Hamburg, Fachbereich Informatik

Zusammenfassung

Die Benutzung eines Web-Browsers ist einfach zu erlernen, dennoch stellt die Navigation im Web selbst für erfahrene Benutzer immer wieder Herausforderungen bereit. Einer der Gründe liegt in der Vielfalt von Links und Linkzielen, die für den Benutzer oft nicht transparent sind und ihn so vor Überraschungen stellen, nachdem er einen Link angeklickt hat. Das Projekt HyperScout beschäftigt sich mit Möglichkeiten, die Navigation zu vereinfachen, indem man Typinformationen von HTML-Links und referenzierten Objekten für den Benutzer sichtbar macht. Dieser Bericht stellt die entwickelten Konzepte und die Ergebnisse einer Evaluation des daraus abgeleiteten Prototyps vor. Die Ergebnisse geben Aufschlüsse darüber, welche Informationen Benutzern vor der Anwahl eines Links hilfreich sind und wie sie dargestellt werden könnten.

1 Einleitung

Obwohl viele Benutzbarkeitsprobleme des Webs bekannt sind, haben sich im Gegensatz zur rasanten technischen Weiterentwicklung die grundlegenden Elemente der Benutzungsschnittstelle von Web-Browsern in den letzten zehn Jahren nur marginal verändert (Bieber 1997). Insbesondere blieb die Schnittstelle von Hyperlinks fast unverändert. Ihre zentrale Bedeutung ergibt sich nicht nur daraus, dass sie aus normalem Text erst Hypertext machen, auch praktisch stellen sie im Web das primäre Mittel für die Navigation zu neuen Informationen dar. Folglich sind sie die wichtigste Interaktionsform im Web: In einer Studie von Catledge und Pitkow waren 52% der Benutzeraktionen Linkaktionen (Catledge 1995), in einer späteren Untersuchung von Tauscher und Greenberg waren es 42% (Tauscher 1997).

Dabei wurden viele Probleme der Navigation in Hypertexten schon lange vor der Entwicklung des Webs charakterisiert. Bereits 1987 definierte Conklin ein Problem mit Hyperlinks, dass er als „Cognitive Overhead“ beschrieb: Lesen von Text und Beachten von Links können sich gegenseitig behindern (Conklin 1987). Der Benutzer wird damit belastet, bei der Navigation seine Aufgabe und bereits aufgenommene Informationen im Blickfeld zu behalten und gleichzeitig Alternativen der Navigation abzuwägen. Diese Problematik verschärft sich, wenn ein Link nicht zu den gewünschten Informationen führt und er zurück navigieren muss, um einen alternativen Weg einzuschlagen. Irrtümlich ausgewählte Links sind damit ein Grund für die bereits 1997 von Tauscher beobachtete „Hub-and-Spoke“-Navigation im Web (Tauscher 1997).

Im Vergleich mit historischen Hypertextsystemen mit typisierten Links, hierarchischen Navigationsmitteln und Link-Karten ist die Schnittstelle der Links im Web vergleichsweise primitiv und wenig aussagekräftig. Ziel des hier beschriebenen Projektes *HyperScout* ist es, eine Erweiterung dieser Benutzungsschnittstelle zu entwickeln und zu testen, ob so die Navigation vereinfacht wer-

den kann. Im Folgenden werden die Probleme von Links genauer betrachtet und Lösungsmöglichkeiten vorgestellt; darauf aufbauend wird das Projekt und ein Prototyp vorgestellt und evaluiert.

2 Typisierte Links und das Web

Die Typisierung von Links ist ein bewährtes Mittel, um die Navigation in Hypertext-Systemen zu vereinfachen. Ein Linktyp beschreibt die Beziehung zwischen Quelle und Ziel eines Links. Er kann damit Benutzern schon vor Anwahl eines Links mehr über seine Bedeutung und das zu erwartende Zielobjekt vermitteln. So kann das Risiko reduziert werden, irrtümlich „falsche“ Links anzuwählen. Aus diesem Grunde führen Thüring et al. „semantic link information“ als erstes Prinzip für benutzbares Hypermedia-Design an (Thüring 1995). Viele Experten aus dem Hypertext-Bereich fordern vor diesem Hintergrund auch für das Web die Umsetzung typisierter Links. Beispielsweise beschreiben Bieber et al. typisierte Links und Knoten als erstes von zehn „high-level hypermedia features“ für die nächste Generation des Webs (Bieber 1997).

Im Web finden sich typisierte Links nur auf den zweiten Blick: Zwar war bereits in Tim Berners-Lees „Proposal“ für das World Wide Web von 1989 eine Typisierung von Knoten und Links vorgesehen (Berners-Lee 1989), aber erst HTML 2.0 von 1995 definiert Linktypen: die beiden Attribute `rel` und `rev` für Anchor-Tags. Sie sollen Links mit einer vorwärts *und* einer rückwärts gerichteten Relation versehen (Berners-Lee 1995). HTML 4.0 definiert hierfür eine Reihe von „nützlichen Typen“ (Ragget 1998), wie beispielsweise für hierarchische Links (`child`, `parent`, `top`) oder Folgen von Seiten (`next`, `previous`). Diese Definitionen haben jedoch kaum praktische Auswirkungen gehabt, da bis heute die gängigen Browser diese Attribute nicht unterstützen.

HTML erlaubt zudem die Angabe von für Menschen lesbaren Bezeichnern für Links über das `title`-Attribut. Dieser Linktitel wird von moderneren Browsern als Tooltip-Popup angezeigt. Einige Web-Guidelines und Usability-Experten (Chisholm 1999; Nielsen 1998) empfehlen die Verwendung dieses Attributes, um die Aussagekraft von Web-Links zu verbessern; dennoch sind sie relativ selten zu finden.

2.1 Grenzen typisierter Links

Obwohl typisierte Links das Potenzial besitzen, die Benutzbarkeit des Webs zu verbessern, gibt es Probleme, die ihre Durchsetzung behindern können.

Eine triviale, aber bedeutende Hürde ist die geringe Verbreitung von Typen im heutigen Web: je mehr Links typisiert sind, desto eher werden Web-Browser Typen unterstützen – und umgekehrt. Es ist aber auch prinzipiell fragwürdig, ob Autoren bereit und in der Lage wären, korrekte Linktypen anzugeben. Frühere Untersuchungen haben wesentliche Autorenprobleme bei der Definition von Typen für Links aufgezeigt. Bei einer zweijährigen Studie mit dem MUCH Hypertext-System wurde von den 200 Benutzern in nur ca. 3% der Fälle ein anderer Linktyp als der Default-Typ angegeben, und selbst in diesen Fällen war die Wahl oft inkonsistent (Wang 1995).

Ein semantisches Problem stellt die Repräsentation von Typen dar. Der von HTML 4.0 verfolgte Ansatz, eine allgemeingültige Typisierung für sämtliche Anwendungsgebiete des Webs zu schaffen, scheint impraktikabel: Bereits bei einer thematisch eingegrenzten Anwendung wies das strikte Typsystem des Hypertext-Systems „Aquanet“ nicht genügend Flexibilität auf und verursachte den Autoren erhebliche Probleme (Marshall 1992). Eine flexible Typisierung, wie sie etwa im Rahmen der „Semantic Web Initiative“ geplant ist, erzeugt Übersetzungsprobleme bei der

Verwendung verschiedener Ontologien. Zudem erfordern aktuelle Standards wie XLink und RDF heute noch einen hohen Mehrfachaufwand bei der Typisierung; neben menschenlesbarer Information ist es möglich, maschinenlesbare Informationen anzugeben, neben der Typisierung von Knoten ist die Typisierung von Relationen möglich.

2.2 Implizite Link- und Objekttypen im Web

Schon aufgrund des Fehlens von Typinformationen erhält der Benutzer im heutigen Web kaum Navigationshilfen. Zusätzlich wird ihm eine Reihe von technisch bereits erreichbaren impliziten Informationen zur Funktion von Links und zu Eigenschaften von Zielobjekten vorenthalten. Derartige Zusatzinformationen lassen sich in sieben Kategorien einteilen:

- **Semantik:** Der Linktext bzw. die verlinkte Grafik geben nur sehr eingeschränkte Informationen über den Inhalt des Zieldokumentes; es wird nicht primär der Inhalt des Zieles beschrieben, sondern die Bedeutung desselben für den Linkautor beim Erstellen des Links. Der Benutzer eines Links kann aber vollkommen andere Zielvorstellungen haben. Begrenzt kann die inhaltliche Bedeutung des Zieldokuments über einfache Verfahren wie die Angabe des Titels oder eine maschinell erzeugte Zusammenfassung des Inhalts angeboten werden.
- **Topologie:** Die Topologie eines Hypertextes bezeichnet die durch seine Links erzeugte Struktur. Dabei lassen sich Dokumenttypen unterschiedlicher Funktion unterscheiden: einige Dokumente dienen der Navigation, andere bieten primär Inhalte an. Links hingegen können der hierarchischen Navigation dienen oder auch Assoziationen ausdrücken. Ein spezieller Aspekt des Webs ist, dass Links auch auf andere Server verweisen können; solche *externen Links* ändern oft in unerwarteter Weise Kontext und Anbieter der Informationen (Spool 1999, S. 45).
- **Benutzung:** Die eigene Benutzung von Dokumenten wird bisher nur vage mittels der Linkfarbe mitgeteilt. Obwohl genauere Informationen über die Navigationsschritte des Benutzers vorliegen (Zeiten, Häufigkeit, Pfade), sind diese nicht zugänglich. Die Benutzung durch Andere, und damit etwa die Popularität der Seiten, ist gar nicht sichtbar.
- **Datei-/ Medientyp:** Verschiedene Medientypen im Web führen zu unerwarteten Auswirkungen: Einige Formate benötigen spezielle Programme oder Plugins zur Darstellung, andere Formate unterstützen keine Links oder ändern die Interaktionsform. Aber selbst HTML-Seiten können unterschiedlichen Standards entsprechen und Darstellungsschwierigkeiten hervorrufen.
- **Aktion:** Zumeist erscheint nach der Auswahl eines Links ein neues Dokument im gleichen Fenster des Browsers. Links können aber auch ganz andere Aktionen bewirken: z.B. auf ein Ziel auf derselben Seite verweisen, ein weiteres Browser-Fenster öffnen oder ein anderes Programm starten (z.B. den E-Mail-Client).
- **Status:** Laut einer Umfrage von 1998 stellen fehlerhafte Links eines der wichtigsten Benutzbarkeitsprobleme im Web dar (GVU 1998). Der Benutzer wird mit kryptischen Fehlermeldungen des Servers konfrontiert und muss sich neu orientieren. Der Anteil von fehlerhaften Links stagniert seit Jahren bei ca. 5% (LinkQuality.com 2003). Es gibt technische Konzepte zur Vermeidung von fehlerhaften Links (Pitkow 1996), die aber praktisch nicht eingesetzt werden.
- **Reaktionszeit:** Die Geschwindigkeit ist ein weiterer zentraler Schwachpunkt des Webs (GVU 1998). Trotz der Zunahme der verfügbaren Bandbreiten (DSL-Techniken, Gigabit-Backbones) kommt es weiterhin zu Engpässen, und akzeptable Reaktionszeiten von 2-4s werden nicht immer erreicht. Die Antwortzeit beim Folgen eines Links ist zwar nicht vorhersagbar, Dateigröße und Antwortzeiten eines Servers geben jedoch Hinweise auf die zu erwartende Reaktivität.

Web-Design-Guidelines und Usability-Experten empfehlen die Kennzeichnung von Links mit unerwarteten Eigenschaften (z.B. zu großen Dateien oder anderen Medientypen), etwa mittels eines kurzen textuellen Hinweises hinter dem Link oder spezieller Icons (Nielsen 1999). Dies bleibt aber dem Autor überlassen und wird, da es zusätzlichen Aufwand bedeutet und bis heute kein Standard existiert, selten umgesetzt.

Die meisten der aufgeführten Informationen lassen sich aus dem Zielobjekt, dem Server, der Benutzung oder dem Link selbst ermitteln und könnten automatisch dargestellt werden. Es stellt sich aber die Frage, *welche* dieser Informationen in welcher Situation hilfreich für den Benutzer sind und *wie* diese Informationen am effizientesten dargestellt werden können.

2.3 Visualisierungsmethoden für Linktypen

Eine Analyse älterer Hypertext-Systeme zeigt eine ganze Reihe von Möglichkeiten auf, um das User-Interface von Links um zusätzliche Informationen anzureichern. Zudem haben auch viele Web-Designer die Problematik der oft missverständlichen Links im Web erkannt, und so finden sich auch hier einige Lösungsansätze:

- **Link-Darstellung:** Unterschiedliche Arten zur Hervorhebung von Links sind eine nahe liegende Methode, um unterschiedliche Typen von Links zu differenzieren. Von Web-Browsern her sind die zwei Link-Farben blau und lila bekannt, es können aber auch unterschiedliche Farbintensitäten, Hintergrundfarben oder Textstile verwendet werden. Ein Vorteil solcher Methoden liegt darin, dass eine entsprechende Codierung kritische Link-Eigenschaften sofort ins Auge fallen lässt. Nachteile liegen in einer möglichen Reduzierung der Lesbarkeit des Textes durch die Hervorhebungen und der begrenzten Aussagevielfalt solcher Methoden.
- **Beim Link:** Im Web wird teilweise versucht, Links durch einen kurzen eingeklammerten Text oder ein Icon hinter dem Link semantisch anzureichern (s.o.). Diese Methode bietet zwar den Vorteil, bereits heute für das Web nutzbar zu sein, derart eingefügten Informationen können aber den Lesefluss stören und sind für viele Layouts und verlinkte Grafiken kaum geeignet.
- **Mauszeiger:** Die Art des Mauszeigers bietet eine weitere etablierte Möglichkeit, dem Benutzer nützliche Informationen zum Systemstatus oder einer mit der Maus auszulösenden Aktion darzustellen; bei Web-Browsern wird sie bisher hingegen kaum verwendet. Ein historisches Beispiel für die Anwendbarkeit der Methode im Hypertext ist das Guide-System, bei dem der Mauszeiger unterschiedliche Browseraktionen ankündigt (Weinreich 2001). Ein Vorteil ist, dass die Information nahe beim Fokus der Aufmerksamkeit dargestellt wird. Andererseits erscheint sie nur, nachdem die Maus über den Link bewegt wurde, und die Möglichkeiten sind recht abstrakt.
- **Popup:** Kleine Popups stellen als so genannte Tooltips inzwischen eine verbreitete Methode dar, um zu bestimmten User-Interface-Elementen kurze, zusätzliche textuelle Hilfen anzubieten. Sie erscheinen aber erst, nachdem man den Mauszeiger kurzzeitig still über ein entsprechendes Element gehalten hat. Popups bieten dafür ebenfalls den Vorteil, beim Fokus der Aufmerksamkeit zu erscheinen.

Es existieren eine ganze Reihe weiterer Möglichkeiten, die hier ausgegrenzt wurden, da sie entweder für Browser ungeeignet erschienen oder kritische Nachteile aufwiesen. Dazu gehört der *reservierte Bereich*, bei dem die zusätzlichen Link-Informationen immer an einer vorgesehenen Stelle erscheinen; beim Web-Browser ist dies die Statusleiste für die URL. Da die Linkdaten aber nicht beim Fokus der Aufmerksamkeit erscheinen, können sie leicht übersehen werden (Zellweger

2000). Ähnlich verhält es sich bei der Verwendung einer *Link-Karte*. Sie wurde von einer ganzen Reihe früherer Hypertext-Systeme verwendet, um einen Überblick zur Struktur des Hypertextes zu geben. Objekte werden dabei durch Symbole dargestellt, Links als Pfeile zwischen ihnen. Systeme wie Aquanet oder Sepia zeigten bei den Pfeilen den Typ der Relation an. Nachteilhaft erscheint, dass die Karte viel zusätzlichen Platz benötigt und der Benutzer die Beziehung zwischen Links in Browser und Karte herstellen muss (Marshall 1992; Thüning 1995).

3 HyperScout

Basierend auf den vorgestellten Überlegungen entstand das Projekt *HyperScout*, in dem Konzepte für eine erweiterte Link-Schnittstelle und deren technische Umsetzung entwickelt wurden. Eine *prototypische Realisierung* sollte im Rahmen dieses Projektes eine experimentelle Untersuchung der Konzepte ermöglichen.

Als Darstellungsmittel für die zusätzlichen Informationen wurden Popups gewählt. Sie bieten für den gewählten Anwendungsbereich die meisten Vorteile: Popups können zum einen direkt beim Link und damit am Fokus der Aufmerksamkeit dargestellt werden, zudem sind sie sehr flexibel und es lassen sich in ihnen beliebige Texte oder Grafiken platzieren. Da die Popups Informationen unterschiedlicher Kategorien anbieten sollten, war ein Entwurfsziel die einfache Unterscheidbarkeit der Kategorien. Zusätzliche Icons am Anfang der Zeilen sollten die Wahrnehmbarkeit vereinfachen, unterschiedliche Hintergrundfarben dienten der Klassifikation und Gruppierung der Informationen. Die Abb. 1, 3, 4 und 5 zeigen Beispiele der verwendeten Popups.



Abb. 1: Ein Link zu einer langen Nachricht in einem News-Ticker

3.1 Das technische Konzept

Ein wesentliches Problem für die Ausgabe von zusätzlichen Linkinformationen liegt darin, dass diese dem Browser bereits *vor* der Anwahl eines Links zugänglich sein müssen. Es wurde ein Konzept benötigt, das auch mit aktuellen Techniken vereinbar ist und die zu übertragende Datenmenge und Netzwerkbelastung nicht wesentlich erhöht. Aus diesem Grunde schied eine rein client-seitige Lösung aus, bei der der Browser alle referenzierten Objekte „pre-fetcht“. Stattdessen setzt HyperScout auf eine server-basierte Lösung, bei der jeder Web-Server die Informationen verwaltet und automatisch zu den Dokumenten hinzufügt. Der Server kann diese Aufgabe beispielsweise mit Hilfe eines Apache-Moduls ausführen, das die übertragenen Seiten parst und um zusätzliche Link-Attribute ergänzt (s. Abb. 2).

```
<A HREF="http://www.scone.de/" HREFLANG="de" HREFTITLE="Scone"
HREFDESC="A Java Framework to Build Web Navigation Tools">
```

Abb. 2: Der HTML-Code eines vom Server ergänzten Anchor-Tags

Das Apache-Modul selbst erhält die Daten aus einer Datenbank, die mittels eines regelmäßig laufenden Robots aktualisiert wird. Sie enthält so nicht nur die Meta-Daten zu lokalen URLs, sondern auch solche zu direkt referenzierten externen Dokumenten (Weinreich 2000). Der Browser kann dann diese Daten auswerten und dem Benutzer z.B. in Form von Popups anbieten. Weitere Informationen wie die Reaktion des Browsers, externe Links oder frühere Aktionen des Benutzers lassen sich aus dem Link-Anker, der URL und der History des Browser ermitteln.

3.2 Der Prototyp

Der zur Evaluation entwickelte Prototyp sollte *möglichst viele* sinnvolle Informationen automatisch extrahieren und auf verständliche Weise präsentieren. Zur Implementierung wurde das Java-Framework Scone verwendet, das die schnelle Realisierung von Prototypen zur Erweiterung der Funktionalität von Browsern und Servern erlaubt (Weinreich 2003).

Der erste Prototyp basierte auf DHTML. Die Popups wurden als unsichtbare <DIV>-Elemente an das Seitenende angefügt und per JavaScript neben dem Mouse-Pointer dargestellt (Weinreich 2000). Leider war diese Lösung für eine längere Evaluation nicht geeignet: Zum einen musste vor Darstellung der Popups die Seite und der umfangreiche zusätzliche DHTML-Code komplett übertragen sein, wodurch inakzeptable Verzögerungen entstanden. Zudem führten Bugs *in den Browsern* dazu, dass sie durch den komplexen DHTML-Code häufig unvorhersehbar abstürzten.

Der zweite Prototyp verwendet ein verstecktes Java Applet, das per JavaScript und LiveConnect über die Benutzeraktionen informiert wird und dann mit Hilfe des Java-AWTs Popup-Fenster beim Mauszeiger darstellt. Dieser Prototyp läuft so stabil, dass längere Evaluationen problemlos durchführbar sind. Zudem bietet diese Methode den Vorteil, dass die darzustellenden Daten erst benötigt werden, wenn das Popup erscheinen soll. Im Hintergrund können die notwendigen Informationen bereitgestellt und bei Bedarf per Socket-Verbindung zum Applet übertragen werden. Der Prototyp verwendet zusätzlich einen Robot, der die notwendigen Informationen client-seitig lädt. So lassen sich Benutzertests mit nahezu beliebigen Web-Sites durchführen.

3.3 Der Benutzertest

Um die Benutzbarkeit des Prototyps und die Nützlichkeit der Konzepte zu evaluieren, wurden sieben Thinking-Aloud-Tests (Lewis 1982) gefolgt von semi-strukturierten Interviews mit jeweils einzelnen Benutzern durchgeführt. Für die Tests wurden im Umgang mit dem Web erfahrene Benutzer gewählt, um Bedienungsprobleme, die nicht auf den Prototyp zurückzuführen sind, zu reduzieren. Die Teilnehmer waren im Durchschnitt 29 Jahre alt, sechs kamen aus dem EDV-Bereich. Die Tests dauerten zwischen 1^h40^{min} und 2^h25^{min}. Als Browser wurde der Internet Explorer 6 unter Windows 2000 eingesetzt, der Rechner war per Ethernet an das Wissenschaftsnetz des DFN angeschlossen. Für den Test wurde (ebenfalls basierend auf Scone) ein Werkzeug zur Unterstützung der Evaluation erstellt, das die Teilnehmer in ein Aufgabenszenario einführte, die Aufgaben auf dem Bildschirm darstellte, sowie sämtliche relevanten Benutzeraktionen aufzeichnete (Weinreich 2003). Die Aufgaben bestanden darin, bestimmte Informationen in sechs vorher ausgewählten Web-Sites unterschiedlichen Charakters zu finden. Dabei wurden die Teilnehmer gebeten, bewusst auf die Popups zu achten. Prätests hatten ergeben, dass sie sonst aus jahrelanger Gewohnheit Links sofort anklickten, so dass die Popups nicht erschienen und somit nicht sinnvoll evaluiert werden konnten.

An dieser Stelle werden primär die im Test ermittelten Probleme und Potenziale hervorgehoben, die konzeptioneller Natur sind und nicht auf Implementierungsdetails des Prototyps beruhen. Bevor die Stärken des Prototyps vorgestellt werden, sollen zunächst die ermittelten Probleme – bezogen auf die angebotenen Inhalte, aber auch auf die Darstellung dieser Inhalte gezeigt werden.

Probleme mit den angezeigten Inhalten

Einige der angezeigten Informationen wurden von den Teilnehmern als zumeist wenig hilfreich eingeordnet. Hierzu gehören topologische Informationen, wie die Art des Verweises bei hierarchischen Links oder die Unterscheidung von Navigations- und Inhaltsseiten. Ebenso wurden viele Detailinformationen zum Zieldokument als überflüssig eingestuft, beispielsweise ob es Formularfelder enthält, andere Medientypen einbindet, oder ob es sich um eine lange Seite handelt. Zur Nützlichkeit von einigen Inhalten gab es geteilte Meinungen, und es bestand mehrfach der Wunsch, die Popups einfach konfigurieren zu können.

Die Menge der angezeigten Daten war offensichtlich häufig deutlich zu groß (Abb. 3). Wir beobachteten während der Tests mehrfach, dass die Teilnehmer nur einige der Informationen in den Popups lasen, andere aber übersahen.

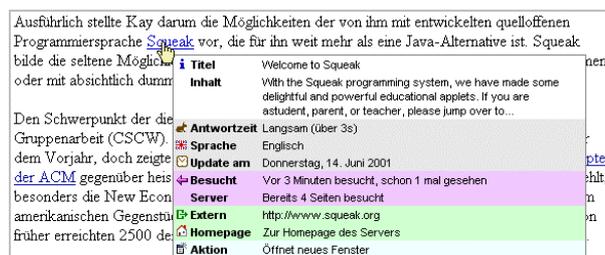


Abb. 3: Ein umfangreiches Popup mit unterschiedlichen Attributtypen

Zuweilen waren die angezeigten Informationen irreführend, beispielsweise wenn der dem Popup zugrunde liegende Titel der Zielseite fehlerhafte Angaben enthielt oder ohne den Seitenkontext missverständlich war. In anderen Fällen zeigten einige Teilnehmer Misstrauen gegenüber korrekten Informationen in den Popups, z.B. wenn der Linktext mit der Zielseite nur schlecht korrespondierte und dadurch der Inhalt des Popups dem Link zu widersprechen schien.

Probleme mit den Popups

Die Testteilnehmer warfen oft nur einen kurzen Blick auf die Popups. Fiel ihnen dabei die gewünschte Information nicht sofort ins Auge, so klickten sie auf den Link. Sie äußerten hierzu, dass die Popups zum Scannen optimiert und wichtige Informationen mehr hervorgehoben werden sollten. Große Popups wurden zudem wiederholt kritisiert, da sie viel verdeckten.

Die gewünschte Darstellung der Informationen variierte erheblich. Oft wurde eine eher visuelle Repräsentation wichtiger Daten gewünscht („Ein Icon für kaputte Links reicht.“), aber einige Teilnehmer wünschten auch mehr technische Angaben, etwa bei fehlerhaften Links.

Alle Teilnehmer haben während der Tests mehrfach auf Links geklickt, ohne auf die Popups zu warten, obwohl sie für den Test explizit darum gebeten wurden. Die Teilnehmer nannten für ihr Verhalten zwei Hauptgründe: Erstens waren sie sich in vielen Fällen sicher, dass der Link „richtig“ sei, zweitens ginge es bei guter Reaktion des Web-Servers oft schneller, direkt das Zieldokument einzusehen, als das Popup abzuwarten. Obgleich so implizit die Wartezeit bis zum Erscheinen der Popups kritisiert wurde, befanden alle Teilnehmer die gewählte Zeitverzögerung (0,8s) als angemessen.

Wir konnten beobachten, dass die Benutzer die Seiten in der Regel erst überflogen und sich schon für einen Link entschieden hatten, *bevor* sie ihn mit der Maus anfuhr. Stellte sich dann aufgrund des Popups der Link als ungeeignet heraus, so war bereits die Mausbewegung vergeblich.

Stärken der erweiterten Link-Schnittstelle

Die angezeigten Informationen waren in vielen Fällen nützlich für die Navigation, da sie den Benutzern bei der Entscheidung halfen, einem Link zu folgen oder nicht. Insgesamt konnten wir beobachten, dass die Popups genauer beachtet wurden, wenn Links missverständlich waren oder die Übertragung der Seiten vom Server lange dauerte. Folgende Informationen zeigten sich als besonders nützlich und wurden von den Testteilnehmern positiv kommentiert:

- Warnungen vor nicht zugreifbaren Dokumenten; hierzu gehören fehlende Dokumente (HTTP Error 404), nicht erreichbare Server und passwort-geschützte Seiten (Abb. 4).
- Die Kennzeichnung externer Links wurde von allen Teilnehmern als sinnvoll angesehen. Einige begrüßten die zusätzliche Angabe des Seitentitels der Homepage des Servers, um den Anbieter der Seiten zu charakterisieren (Abb. 4).



Abb. 4: Ein externer Link zu einer nicht mehr existierenden Seite

- Die inhaltlichen Angaben halfen, sofern sie den Linktext relevant ergänzten. Gelobt wurde insbesondere die aus dem „Description“-Tag der Zielseite gewonnene Beschreibung (Abb. 1).
- Kleine Popups wurden mehrfach als „optimal“ bezeichnet, z.B. solche, die nur eine E-Mail-Adresse oder den Verweis an eine andere Stelle desselben Dokumentes beschrieben (Abb. 5).

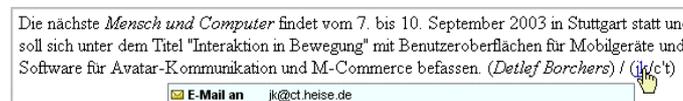


Abb. 5: Ein kleines Popup mit einer E-Mail-Adresse

- Die Angabe, dass der Browser ein neues Fenster öffnen würde, wurde gelobt. Gleichzeitig kritisierte man aber, dass sich dieses oft unerwünschte Verhalten nicht unterdrücken ließe.
- Die Informationen zur eigenen Benutzung scheinen hilfreich. Zwei Tester betonten, dass die Benutzungsinformation des Browsers in Form von lila Links sonst oft zu vage ist bzw. bei Grafiken ganz fehle.
- Eine Reihe von Informationen wurde nur dann begrüßt, wenn sie auf *unerwartete Eigenschaften* des Zieles hinwiesen, beispielsweise nicht vom Browser darstellbare Dateitypen, alte Seiten, nicht beherrschte Sprachen oder sehr große Dateien.

Als positive Eigenschaft der Popups stellte sich heraus, dass sie Informationen „On Demand“ boten. Dadurch haben sie weder das normale Erscheinungsbild der Seiten verändert noch beim Lesen abgelenkt, sondern erweiterten die Möglichkeiten des Browsers lediglich. Gelobt wurde zudem, dass die Popups dem Web ein „einheitlicheres“ User Interface gäben.

Alle Teilnehmer äußerten, dass ihnen der Umgang mit dem durch HyperScout erweiterten Browser „Spaß“ gemacht hätte, wobei drei Teilnehmer es bereits in der getesteten Form gerne in ihrem Browser integriert hätten.

3.4 Diskussion und Ausblick

Die Tests haben zwei Kernprobleme des Prototyps offenbart: Erstens müssen die Benutzer die Maus zum Link bewegen und auf das Erscheinen des Popups warten, um die zusätzlichen Informationen sehen zu können. Bei einigen kritischen Linkeigenschaften, wie beispielsweise fehlerhaften Links, ist dies zu aufwändig. Eine unmittelbarere Art der Darstellung wäre geeigneter.

Zweitens waren die dargestellten Informationen oft zu umfangreich, als dass sie mit einem Blick hätten erfasst werden können. Dies liegt zum Teil an dem verfolgten Forschungsziel, möglichst viele Informationen auf Ihre Nützlichkeit hin zu untersuchen. Benutzer scheinen im Umgang mit dem Web aber das *Scannen* von Informationen gewohnt zu sein (Morkes 1997). Aus diesem Grunde sollten neben einer Reduzierung der Informationen die Popups auch für die rasche Wahrnehmung optimiert werden und wichtige Informationen stärker hervorheben. Für textuelle Informationen wie einer Inhaltsbeschreibung scheint die Darstellung angemessen, aber für abstrahierbare Attribute, z.B. fehlerhaften Verweisen, böte sich eine grafische Darstellung an.

Eine Alternative zum getesteten Prototyp stellt die Möglichkeit dar, mehrere Detailstufen für zusätzliche Link-Informationen anzubieten. Kritische Hinweise, wie auf fehlerhafte Links, sollten schon in der Art der Hervorhebung des Links offenbar werden. Um die Darstellung der Seiten dabei möglichst wenig zu beeinträchtigen, böten sich beispielsweise durchscheinende farbige Flächen („Overlays“) an, welche alle Links entsprechend markieren. Die „Link-On-Demand“-Technik ist zudem eine geeignete Option, um die Seite zum Lesen unverändert darzustellen, die Overlays dann für die Navigation z.B. per Tastendruck erscheinen zu lassen (Weinreich 2001).

Als zweite Detailstufe könnte der Mauszeiger eingesetzt werden, um weitere Link-Informationen ohne Wartezeit anbieten zu können. Hierzu zählen beispielsweise externe Links, Verweise zu Homepages und Links, die neue Browserfenster öffnen.

4 Resümee

Die positive Aufnahme des Prototyps weist darauf hin, dass sich die Benutzungsschnittstelle von Links im Web mit relativ einfachen Mitteln deutlich verbessern lässt. Die Benutzertests zeigten aber auch, dass Verbesserungsbedarf bei der Auswahl und Darstellung der Informationen besteht: Die Inhalte der Popups müssen auf wesentliche Daten reduziert werden und die Darstellung der Link-Attribute muss auch andere, unmittelbar sichtbare Methoden verwenden.

Mit Hilfe der hier vorgestellten Konzepte kann Benutzern über Web-Sites hinweg eine konsistente, erweiterte Schnittstelle für Links angeboten werden, die mehr Transparenz und Sicherheit bei der Navigation erlaubt. Dabei wird nicht die Gestaltungsfreiheit der Autoren eingeschränkt sondern ihre Arbeit erleichtert: Anstatt kritische Links selbst kennzeichnen zu müssen, wird dies durch die Technik unterstützt. Zudem kann mit Hilfe der Konzepte von *HyperScout* die URL als Benutzungsschnittstelle mehr in den Hintergrund rücken. Diese wenig ergonomische Informationsdarbietung verliert sukzessive an Aussagekraft, da zunehmend verwendete Content-Management-Systeme hier lediglich IDs anbieten.

Zurzeit wird ein weiterer Prototyp entwickelt, in den die Erkenntnisse der Tests einfließen.

Literaturverzeichnis

- Berners-Lee, T. (1989): *Information Management: A Proposal*. Graz: CERN Internal Communication.
- Berners-Lee, T.; Connolly, D. (1995): HTML 2.0 Specification. W3C: <http://www.w3.org/MarkUp/html-spec/>
- Bieber, M.; Vitali, F.; Ashman, H.; Balasubramanian, V.; Oinas-Kukkonen, H. (1997): Fourth generation hypermedia: some missing links for the World Wide Web. *Int. J. Human-Computer Studies*, 47(1): 31-65
- Catledge, L. D.; Pitkow, J. E. (1995): *Characterizing Browsing Strategies in the World-Wide Web*. *Computer Networks and ISDN Systems*, 27(6): 1065-1073.
- Chisholm, W. et al. (1999): *Web Content Accessibility Guidelines*, W3C: <http://w3.org/TR/WAI-WEBCONTENT/>
- Conklin, J. (1987): *Hypertext: An Introduction and Survey*. *IEEE Computer* 20(9): 17-41.
- GVU: Graphic, Visualization, and Usability Center (1998): *10th WWW User Survey*, Gvu Center, College of Computing, Georgia Institute of Technology, Atlanta: http://www.cc.gatech.edu/gvu/user_surveys/
- Lewis, C. (1982): *Using the "Thinking-aloud" Method in Cognitive Interfaces Design*. Yorktown Heights, NY, IBM Thomas J. Watson Research Center, Report RC 9265 (#40713)
- LinkQuality.com (2003): *The Web's Integrity Benchmark*, Linkalarm Inc.: <http://linkquality.com/>
- Marshall, C. C.; Rogers, R. A. (1992): *Two Years before the Mist: Experiences with Aqanet*. ACM Press: European Conference on Hypertext Technology (ECHT '92), Milano, Italy: 53-62.
- Morkes, J.; Nielsen, J. (1997): *Concise, SCANNABLE, and Objective: How to Write for the Web*. Sun Microsystems: <http://www.useit.com/papers/webwriting/writing.html>
- Nielsen, J. (1998): *Using Link Titles to Help Users Predict Where They Are Going*, Jakob Nielsen's Alertbox.
- Nielsen, J. (1999): *Designing Web Usability: The Practice of Simplicity*, New Riders Publishing.
- Pitkow, J. E.; Jones, R.K. (1996): *Supporting the Web: A Distributed Hyperlink Database System*. *Computer Networks and ISDN Systems* 28(7-11): 981- 991.
- Raggett, D.; Le Hors, A.; Jacobs, I. (1998): HTML 4.0 specification. W3C: <http://w3.org/TR/REC-html40>
- Spool, J. et al. (1999): *Web Site Usability: A Designer's Guide*. Morgan Kaufmann Publishers.
- Tauscher, L.; Greenberg S. (1997): *How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems*. *Int. J. Human-Computer Studies* 47(1): 97-137.
- Thüring, M.; Hannemann, J.; Haake, J. (1995): *Hypermedia and Cognition: Designing for Comprehension*. *ACM Communications*, 38(8): 57-66.
- Wang, W.; Rada, R. (1995): *Experiences with Semantic Net Based Hypermedia*. *Int. Journal of Human Computer Studies* 43(3): 419-439.
- Weinreich, H.; Buchmann, V.; Lamersdorf W. (2003): *Scone: Ein Framework zur evaluativen Realisierung von Erweiterungen des Webs*. Springer: Tagungsband zur KiVS 2003: 31-42.
- Weinreich, H.; Lamersdorf, W. (2000): *Concepts for Improved Visualization of Web Link Attributes*. *Computer Networks* 33(1-6): 403-416.
- Weinreich, H.; Obendorf H.; Lamersdorf, W. (2001): *The Look of the Link - Concepts for the User Interface of Extended Hyperlinks*. ACM Press: 12th Conference on Hypertext, Århus, Denmark: 19-28.
- Zellweger, P.; Regli, S.; Mackinlay, J.; Chang B-W. (2000): *The Impact of Fluid Documents on Reading and Browsing: An Observational Study*. ACM Press: CHI 2000 Conference, Den Hague, NL: 249-256.

Kontakt

Harald Weinreich*, Hartmut Obendorf^x, Winfried Lamersdorf*

E-Mail: {weinreich | obendorf | lamersdorf }@informatik.uni-hamburg.de

*AG Verteilte Systeme und Informationssysteme, ^xAngewandte und Sozialorientierte Informatik

Fachbereich Informatik, Universität Hamburg: <http://www.informatik.uni-hamburg.de/>