

Hin und her im Hyperspace

Verlustloses lineares Navigieren

Martin Rotard, Matthias Ressel, Thomas Ertl

Universität Stuttgart, Fakultät Informatik

Institut für Visualisierung und Interaktive Systeme

E-Mail: {rotard, ressel, ertl}@informatik.uni-stuttgart.de

Kurzfassung

Im Internet zu browsen gehört inzwischen zur Lieblingsbeschäftigung vieler Menschen. Die Unterstützung der Navigation spielt dabei eine wichtige Rolle. Benutzer müssen sich effektiv und effizient in der näheren Umgebung der gerade angezeigten Seite hin- und herbewegen können. Das hierzu heute am meisten eingesetzte Verfahren des linearen Browsens ist verlustbehaftet. Dieser Beitrag stellt daher alternative neuartige Browsing-Verfahren vor und arbeitet deren Vor- und Nachteile heraus.

1 Einleitung

Das Internet ist aus der heutigen Welt nicht mehr wegzudenken. Das World Wide Web ist dabei eine sehr erfolgreiche Umsetzung des Hypertext-Konzepts zum Browsen von Informationen. Im Hypertext sind Informationen nicht linear strukturiert und repräsentiert, sondern durch organisatorische oder assoziative Beziehungen verknüpft. Die daraus resultierenden Benutzerprobleme sind Desorientierung und kognitiver Mehraufwand (vgl. "lost in hyperspace" [2]). Zum einen verliert der Benutzer das Gefühl für Ort und Richtung durch die vielen Freiheitsgrade und Dimensionen zur Erkundung von Informationsstrukturen. Zum andern stellt das Auswählen und Verfolgen von Verweisen neben der eigentlichen Problemlösung einen zusätzlichen kognitiven Aufwand des Konzentrations- und Erinnerungsvermögens dar. Jeder Benutzer schafft sich einen persönlichen Informationsraum durch das Aufrufen von Seiten. Der persönlichen Informationsraum geht aber verloren, wenn keine geeigneten Mechanismen vorhanden sind, um diesen zu sichern [1, 9]. Dies ist die Aufgabe der Navigationsmechanismen eines Hypertext-Browsers.

Der Benutzer benötigt Unterstützung, um gerade besuchte Seiten wieder aufzurufen. Dafür stellen alle gängigen Browser Navigationsmechnismen in Form von Vorwärts- und Zurück-Buttons zur Verfügung (Abbildung 1). Dieser an sich einfach zu bedienende Mechanismus führt häufig – und für den Benutzer zudem überraschend – dazu, dass der Zugriff auf bestimmte Seiten über die Navigationsbuttons verloren geht. Dieser Beitrag beschreibt dieses Problem und stellt alternative, zum Teil völlig neuartige Verfahren vor, die verlustfrei arbeiten, ohne auf die Vorteile der einfachen Bedienbarkeit zu verzichten.



Abbildung 1: Würden Sie diesen Navigationsbuttons vertrauen?

Daneben gibt es viele Ansätze für die graphische Modellierung des Informationsraums. Die Komplexität der vielen graphischen Informationen wird durch dynamische Generierung der nahen Umgebung des Informationsraums reduziert [3, 4]. In keinem der heute gängigen Hypertext-Browser ist ein solcher Ansatz integriert. Dies zeigt, dass sich diese Visualisierungen nicht durchgesetzt haben.

In unserem Beitrag verwenden wir den Begriff *Navigationshistorie*. Damit ist das Modell gemeint, das sich hinter den Navigationsbuttons verbirgt. Die Navigationshistorie unterscheidet sich wesentlich von der in Browsern anzeigbaren *Historie-Liste*, die alle Adressen von Seiten speichert, auf die innerhalb eines bestimmten Zeitraums, z. B. innerhalb von 30 Tagen, zugegriffen wurde. In der Historie-Liste kommt jede Seite maximal einmal

vor, Vorgänger-Nachfolger-Relationen bleiben hierbei unbeachtet. Die Historie-Liste im Browser besitzt globale Gültigkeit über alle Browserfenster hinweg. Jedes Browserfenster verwaltet dagegen eine eigene Navigationshistorie.

In Kapitel 2 werden einige herkömmliche Browsing-Verfahren beschrieben, an Beispielen illustriert und anschließend jeweils kurz bewertet. Kapitel 3 stellt in gleicher Weise drei neuartige Verfahren vor. Eine Umsetzung der neuen Algorithmen wird in Kapitel 4 beschrieben. Die Verfahren werden in Kapitel 5 einem kritischen Vergleich unterzogen. Abschließend gibt Kapitel 6 einen Ausblick auf die möglichen Einsatzbereiche.

2 Herkömmliche Browsing-Verfahren

Dieses Kapitel beschreibt mehrere bekannte Browsing-Verfahren [11], u. a. die weit verbreiteten Linearen Browsing-Verfahren. Wir werden diese mit Hilfe von Diagrammen und der folgender Terminologie einführen: S_i bezeichne aufgerufene Seiten und t_i chronologische Zeitpunkte mit $t_i < t_{i+1}$. $S(t_i)$ bezeichne die Seite, die zum Zeitpunkt t_i aufgerufen wird. Im Diagramm zeigt ein Pfeil von S_m nach S_n , dass man mit Hilfe des Vorwärts-Buttons von S_m nach S_n bzw. mit Hilfe des Zurück-Buttons von S_n nach S_m navigieren kann.

S_n heißt direkter Nachfolger von S_m , falls S_n von S_m aus aufgerufen wurde, sei es durch Verfolgen eines Links, Eintippen einer URL oder Auswahl eines Lesezeichens. Formal gibt es dann aufeinanderfolgende Zeitpunkte t_i und t_{i+1} , so dass $S_m = S(t_i)$ und $S_n = S(t_{i+1})$ gilt. Analog heißt dann S_m direkter Vorgänger von S_n . S_m und S_n nennen wir dann auch direkte Nachbarn.

Normalerweise erwarten wir, dass in der Navigationshistorie nur dann ein Pfeil von S_m nach S_n zeigt, wenn S_n direkter Nachfolger von S_m ist. Wir nennen Verfahren, die dies einhalten, nachbarschaftserhaltend, da man durch Betätigen des Vorwärts- und Zurück-Buttons nur zu verwandten Nachbarseiten kommt. Dies ist keine Selbstverständlichkeit, wir werden im folgenden ein Verfahren sehen, welches nicht nachbarschaftserhaltend ist.

2.1 Lineare Browsing-Verfahren

Idee: Das Lineare Browsing-Verfahren (Linear His-

tory List in [11]) findet in allen gängigen Hypertext-Browsern ihren Einsatz. Es leitet seinen Namen vom linearen Aufbau der Navigationshistorie ab. Die Bedienung ist für den Benutzer einfach. Er benötigt nur zwei Buttons (Abbildung 1) - einen, um in der Navigationshistorie zurück und einen weiteren, um darin vorwärts zu navigieren.

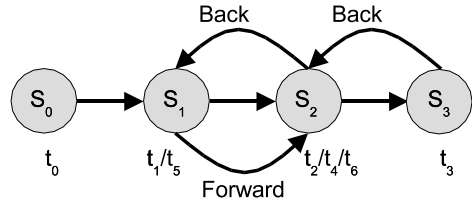


Abbildung 2: Navigation in der Historie

Beschreibung: Wenn eine neue Seite aufgerufen wird und die aktuelle Seite die letzte Seite der Navigationshistorie ist, wird die neue Seite am Ende angehängt. Ist die aktuelle Seite nicht die letzte Seite, werden zuvor alle Folgeseiten der aktuellen Seite eliminiert. Dies ist notwendig, um die Linearität der Navigationshistorie weiter zu wahren. Der Button Zurück ruft die Vorgängerseite der aktuellen Seite auf, der Button Vorwärts den Nachfolger, jeweils sofern vorhanden. Dabei werden keine Seiten an die Navigationshistorie angehängt, sondern die Navigationshistorie wird nur traversiert.

Beispiel: In Abbildung 2 verwendet der Benutzer nach einer Folge von Seitenaufrufen S_0 bis S_3 zwei mal den Zurück-Button. Dadurch werden ihm die vorherigen Seiten S_2 und S_1 angezeigt. Anschließend betätigt er einmal den Vorwärts-Button und gelangt dadurch wieder zu Seite S_2 . Es gilt also $S(t_0)=S_0$, $S(t_1)=S(t_5)=S_1$, $S(t_2)=S(t_4)=S(t_6)=S_2$ und $S(t_3)=S_3$.

Ruft er allerdings nach der Zurück-Sequenz von Seite S_1 aus eine neue Seite auf (Zeitpunkt t_5 in Abbildung 3), werden die bisherigen Folgeseiten S_2 und S_3 aus der Navigationshistorie eliminiert. Anschließend sind für den Benutzer nur noch die Seiten S_0 , S_1 und S_4 erreichbar.

Bewertung: Die Linearität dieses Verfahrens fördert die Überschaubarkeit für den Benutzer bei der Navigation, da sie als Interaktionselemente nur zwei Buttons benötigt, deren Funktionsweise eindeutig

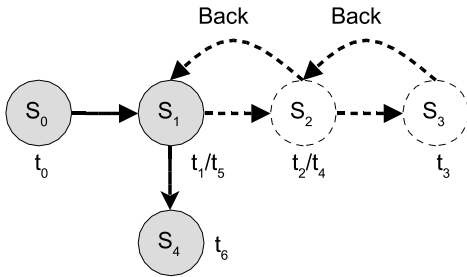


Abbildung 3: Der Seitenaufruf S_4 führt zum Verlust des Zugriffs auf die Seiten S_2 und S_3

erscheint. Der Benutzer kann sich somit zunächst ein klares mentales Modell aufbauen [10]. Allerdings kann es, wie im Beispiel gezeigt, zu einem Informationsverlust innerhalb der Navigationshistorie kommen, ohne dass der Benutzer das in diesem Moment erwartet oder gar sofort bemerken könnte. Das mentale Modell stimmt nun nicht mehr mit dem Modell des Browsers überein. Der Benutzer hat auch keine Möglichkeit, den Vorgang zu stornieren. Falls der Benutzer auf diese Seiten wieder zugreifen will, muss er im obigen Beispiel die Seiten S_2 und S_3 erneut manuell aufrufen. Dabei gehen dem Benutzer in der Navigationshistorie aber wieder der Zugriff auf Seiten verloren, diesmal auf die Seite S_4 .

2.2 Unstetig-lineares Verfahren

Idee: Dieses Verfahren ist eine Erweiterung des Linearen Browsing-Verfahrens, die verhindert, dass in der Navigationshistorie der Zugriff auf Seiten verloren geht.

Beschreibung: Wenn eine Seite direkt - über Verfolgen eines Links oder URL-Eingabe - aufgerufen wird, dann wird sie ans Ende der Navigationshistorie angehängt (Abbildung 4). Die Funktionsweise der Navigation über die Zurück- und Vorwärts-Buttons entspricht dem Linearen Browsing-Verfahren.

Beispiel: Mit denselben Benutzeraktionen wie im Beispiel aus Abschnitt 2.1, würde die Seite S_4 ans Ende, also an S_3 angehängt (vgl. Abbildung 4). Zwischen S_3 und S_4 besteht aber keinerlei nachbarschaftliche Beziehung.

Bewertung: Der Benutzer kann über die Zurück- und Vorwärts-Buttons alle Seiten der Navigations-

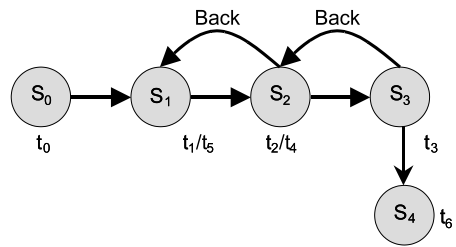


Abbildung 4: Seitenfolge des Unstetig-linearen Verfahrens

historie erreichen. Das Unstetig-lineares-Verfahren ist dabei nicht nachbarschaftserhaltend.

2.3 Travel-Browsing-Verfahren

Idee: Das Travel-Browsing-Verfahren (Branching History List in [11] bzw. Hierarchical History List in [7]) ist eine Erweiterung des Linearen Browsing-Verfahrens, welches den Informationsverlust der Navigationshistorie vermeidet und zusätzlich die Nachbarschaftsbeziehungen erhält. Hierzu wird eine baumförmige Navigationshistorie eingesetzt, bei der eine Seite mehrere Folgeseiten haben kann.

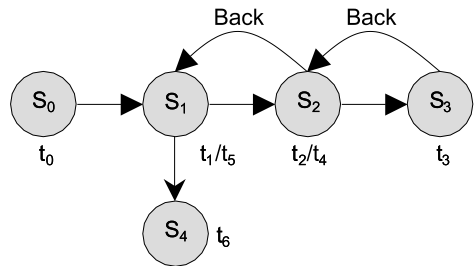


Abbildung 5: Baumförmige Seitenfolge des Travel-Browsing-Verfahrens

Beschreibung: Die Funktionsweise des Travel-Browsing-Verfahrens entspricht dem Linearen Browsing-Verfahren, mit folgenden Änderungen: Wenn eine neue Seite aufgerufen wird und die aktuelle Seite nicht die letzte Seite einer Seitenfolge in der Navigationshistorie ist, wird die neue Seite als zusätzliche Folgeseite der aktuellen Seite definiert.

Wenn der Benutzer von einer Seite mit mehreren Folgeseiten aus Vorwärts navigiert, dann muss er sich entscheiden, zu welcher Folgeseite er wechseln möchte.

Beispiel: Bis zum Zeitpunkt t_5 in Abbildung 5 unterscheidet sich das Lineare vom Travel-Browsing-Verfahren für den Benutzer nicht. Ruft der Benutzer dann die neue Seite S_4 auf, wird diese als zusätzliche Folgeseite für S_1 eingefügt. Die Navigationshistorie verzweigt bei Seite S_1 baumförmig, d. h. diese Seite besitzt zwei Folgeseiten S_2 und S_4 . Navigiert der Benutzer von Seite S_4 aus mit dem Zurück-Button auf Seite S_1 und betätigt dort den Vorwärts-Button, dann muss er sich zwischen den beiden Folgeseiten S_2 und S_4 entscheiden. Dies kann beispielsweise über ein Auswahlmü erfolgen.

Bewertung: Beim Travel-Browsing-Verfahren steht die verlustlose Navigation im Vordergrund. Dadurch kann der Benutzer alle bereits angezeigten Seiten erneut aufrufen. Die baumförmig verzweigte Navigationshistorie erfordert dafür allerdings ein zusätzliches Auswahlmü, falls mehrere Folgeseite zur Auswahl stehen. Die kognitive Belastung ist dadurch wesentlich höher, als beim Linearen Browsing-Verfahren. Ein einfaches und effizientes Bedienen über lediglich zwei Buttons ist nicht mehr möglich.

3 Verlustloses lineares Navigieren

In diesem Kapitel werden drei neuartige verlustlose Browsing-Verfahren vorgestellt. Die in Kapitel 2 eingeführte Terminologie wird um die Darstellung erweitert, eine markierte Seite mit einem doppelten Rand zu kennzeichnen. Diese Markierung stellt ein Seitenattribut dar, das für die algorithmische Weiterverarbeitung benötigt wird.

3.1 Funktionsweise des verlustlosen linearen Navigierens

Idee: Das Verfahren des verlustlosen linearen Navigierens ist eine Kombination des Linearen und des Travel-Browsing-Verfahrens. Dieses Verfahren vermeidet den Verlust von Seiten, ohne zusätzliche Entscheidungen des Benutzers zu erfordern.

Beschreibung: Die Funktionsweise hat nur eine Änderung zum Linearen Browsing-Verfahren: Wenn eine neue Seite aufgerufen wird und die aktu-

elle Seite nicht die letzte Seite der Navigationshistorie ist, werden alle Seiten von der aktuellen Seite bis zur vorletzten Seite der Navigationshistorie in umgekehrter Reihenfolge am Ende hinzugefügt. Anschließend wird die neue Seite angehängt.

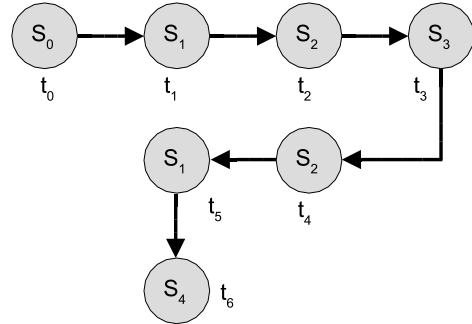


Abbildung 6: Navigationshistorie des verlustlosen linearen Navigierens

Beispiel: Bei diesem Verfahren bemerkt der Benutzer bis zum Zeitpunkt t_5 in Abbildung 3 keinen Unterschied im Verhalten. Beim Aufruf der neuen Seite S_4 werden alle Seiten von der aktuellen Seite S_1 bis zur vorletzten Seite S_2 der Navigationshistorie in umgekehrter, sozusagen gespiegelter Reihenfolge am Ende eingefügt. Danach wird noch die neue Seite S_4 angehängt. In Abbildung 6 ist die durch diese Umformungen entstehende Navigationshistorie dargestellt. Alle Seiten in der Vergangenheit sind damit wieder für den Benutzer erreichbar. Wenn der Benutzer von Seite S_4 aus zurück navigiert, ergibt dies die Seitenfolge S_4 - S_1 - S_2 - S_3 - S_2 - S_1 - S_0 im Gegensatz zu S_4 - S_1 - S_0 beim Linearen Browsing-Verfahren (vergleiche Abbildung 3).

Bewertung: Das Verfahren des verlustlosen linearen Navigierens hat ein klares Konzept und kommt ohne Seitenverlust, ohne Sonderfälle und ohne zusätzliche Auswahldialoge aus. Es kombiniert für den Benutzer alle Vorteile der in Kapitel 2 beschriebenen Verfahren, vermeidet aber deren Nachteile. Jedoch ist durch das verlustfreie Verfahren die Seitenfolge beim Zugriff auf ältere Seiten länger, da der Benutzer im Beispiel etwa den gesamten Zyklus S_1 - S_2 - S_3 - S_2 - S_1 zurück navigieren muss, um zu S_0 zu gelangen. Der Benutzer muss sozusagen einen Umweg gehen.

3.2 Erweiterungen des verlustlosen linearen Navigierens

Beim verlustlosen linearen Navigieren muss der Benutzer u. U. erst einen Zyklus von Seiten vollständig durchlaufen, um auf ältere Seiten zugreifen zu können. In diesem Abschnitt werden erweiterte Verfahren beschrieben, die dem Benutzer die Möglichkeit geben, diese Zyklen zu überspringen.

3.2.1 Hyper-Back

Idee: In der Navigationshistorie entstehen beim Verfahren des verlustlosen linearen Navigierens Zyklen. Diese kann der Benutzer durch Verwenden des Hyper-Back-Buttons (Abbildung 7) überspringen.



Abbildung 7: Der Hyper-Back-Button (ganz links) ermöglicht das Überspringen von Zyklen

Beschreibung: Für dieses Verfahren wird die Funktionsweise des verlustlosen linearen Navigierens um einige Punkte erweitert: Wenn eine neue Seite aufgerufen wird und die aktuelle Seite nicht die letzte Seite der Navigationshistorie ist, werden alle Seiten von der aktuellen Seite bis zur vorletzten Seite der Navigationshistorie in umgekehrter Reihenfolge am Ende hinzugefügt. Zusätzlich wird in der daraus entstandenen Navigationshistorie die letzte Seite als Zyklusende speziell markiert und anschließend wird die neue Seite daran angehängt.

Der neu hinzu gekommene Hyperback-Button wird genau dann aktiviert, wenn es möglich ist einen Zyklus zu überspringen, d. h. wenn die aktuelle Seite eine Markierung besitzt. Wird dieser Button vom Benutzer verwendet, sucht ein einfacher Algorithmus rückwärts in der Navigationshistorie nach dem ersten weiteren unmarkierten Auftreten dieser Seite. Dem Benutzer wird dann die Seite davor als neue aktuelle Seite angezeigt. Damit wird nicht nur der letzte Zyklus, sondern auch alle anderen Zyklen übersprungen, die von der markierten Seite aus begonnen haben.

Der Vorwärts- und Rückwärts-Button hat die selbe Funktion wie beim Verfahren des verlustlosen linearen Navigierens. Jedoch muss bei der Zurück-

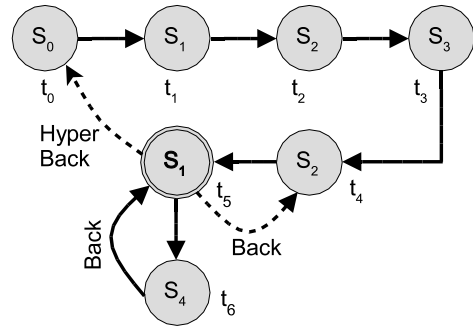


Abbildung 8: Hyper Back ermöglicht das Überspringen eines Zyklus in der Navigationshistorie

und Hyper-Back-Navigation zusätzlich protokolliert werden, welche Seitenfolge der Benutzer rückwärts navigiert hat. Dadurch ist es wieder möglich, genau diese Seitenfolge vorwärts zu navigieren. Bei jedem Vorwärts-Schritt wird die zuletzt protokollierte Seite aus dem Protokoll entfernt, damit der Benutzer jederzeit von der neuen aktuellen Seite aus wieder per Zurück- oder Hyper-Back-Button zurück navigieren kann.

Beispiel: S_4 ist die aktuelle Seite, die der Browser gerade anzeigt (Abbildung 8). Der Benutzer verwendet den Zurück-Button um Seite S_1 aufzurufen. Da diese Seite eine Markierung enthält, wird der Hyper-Back-Button aktiviert. Wenn der Benutzer diesen Button drückt, wird als nächstes die Seite S_0 angezeigt. Wenn er dagegen den Zurück-Button verwendet, kehrt er wie gewohnt zur Seite S_2 zurück.

Bewertung: Der Hyper-Back-Button erweitert die Vorteile des verlustlosen linearen Navigierens. Der Benutzer kann dadurch Zyklen überspringen und damit schneller auf ältere Seite zugreifen.

3.2.2 Trace-Back

Idee: Eine Kombination des Linearen Browsing-Verfahrens und des verlustlosen linearen Navigierens ist das Trace-Back-Verfahren. Dabei wird der Trace-Back-Button (Abbildung 9) verwendet, um einen Zyklus (rückwärts) zu betreten.

Beschreibung: Die Funktionsweise dieses Verfahrens ist analog zum Hyper-Back-Verfahren in Kapitel 3.2.1, aber mit folgenden Änderungen: Der



Abbildung 9: Navigationsbuttons mit dem Trace-Back-Button in der Mitte

Trace-Back-Button ist dann aktiv, wenn die aktuelle Seite als Zyklusende markiert ist. Bei der Betätigung dieses Buttons wird der Zyklus rückwärts betreten d. h. der Vorgänger der aktuellen Seite in der Navigationshistorie aufgerufen. Dagegen überspringt der Zurück-Button den Zyklus (analog zum Hyper-Back-Button) bei einer markierten Seite. Durch diese Wahlmöglichkeit der beiden Rückwärts-Pfade, müssen alle Trace-Back- und Zurück-Schritte protokolliert werden, um die konsistente Vorwärts-Navigation zu ermöglichen.

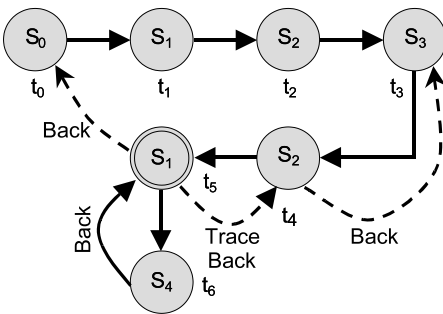


Abbildung 10: Trace Back ermöglicht das Betreten eines Zyklus in der Navigationshistorie

Beispiel: Der Benutzer zeigt gerade die Seite S_4 an (Abbildung 10). Anschließend verwendet er den Zurück-Button, um die zuletzt angezeigte Seite S_1 aufzurufen. Da diese Seite eine Markierung trägt, wird der Trace-Back-Button aktiviert. Die Verwendung dieses Buttons zeigt die Seite S_2 an und betritt damit den Zyklus. Da die Seite S_2 nicht mehr markiert ist, wird der Button deaktiviert. Verwendet der Benutzer dagegen den Zurück-Button bei Seite S_1 , dann wird entsprechend dem linearen Verfahren die Seite S_0 aufgerufen.

Bewertung: Die Kombination des Linearen Browsing-Verfahrens und des verlustlosen linearen Navigierens ermöglicht dem Benutzer das selektive

Betretten von Zyklen mit dem Trace-Back-Button. Dagegen stimmt die Navigation per Zurück-Button mit der des Linearen Browsing-Verfahrens überein.

4 Implementierung

Die in Kapitel 3 vorgestellten Browsing-Verfahren wurden alle implementiert und in einen selbstentwickelten Browser integriert. Dazu wurde die für Java verfügbare CalHTMLPane [6] verwendet, die es ermöglicht, HTML 3.2/4.0 anzuzeigen. Diesen Browser verwenden wir auch, um die Verfahren im täglichen Einsatz auf ihre Gebrauchstauglichkeit zu testen. Wir gingen von der Hypothese aus, dass sich Benutzer nur für die nahe zeitliche Historie ein mentales Modell der besuchten Seiten aufbauen. Wir vermuteten, dass die Verfahren des verlustlosen linearen Navigierens dies am besten unterstützen. In der Tat konnten unsere ersten praktischen Versuche dies bestätigen.

Um für den Benutzer eine noch bessere Orientierung zu gewährleisten, haben wir in unserem Browser für jeden Navigationsbutton ToolTips entwickelt, die dem Benutzer schon vorab den Titel der Seite anzeigen, die aufgerufen wird, wenn er diesen Button betätigt (Abbildung 11).

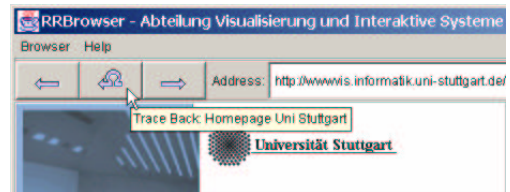


Abbildung 11: ToolTips zeigen den Titel der Seite an, die mit dem Button aufgerufen wird

5 Kritischer Vergleich

Abbildung 12 vergleicht die vorgestellten Verfahren bezüglich verschiedener Kriterien. Nur *Verlustfreiheit* garantiert dem Benutzer, dass er über die Navigationsbuttons alle besuchten Seiten wieder erreicht. *Linearität* unterstützt die einfache Bedienbarkeit über zwei Buttons, ohne vom Benutzer zusätzliche Entscheidungen abzuverlangen. Das Erhalten der *Nachbarschaftsbeziehung* erleichtert Be-

Verfahren	Linear	Travel	Unstetig-linear	Verlustlos lineares Navigieren	Hyper-Back	Trace-Back
verlustfrei	-	+	+	+	+	+
linear	+	-	+	+	o	o
nachbarschafts-erhaltend	+	+	-	+	+	+
effizient in naher Vergangenheit	o	+	o	+	+	+
effizient in älterer Vergangenheit	o	+	+	-	+	+

Abbildung 12: Vergleich der Browsing-Strategien

nutzern die Orientierung. Ein Verfahren ist *effizient*, wenn es keine unnötigen Interaktionsschritte benötigt, um zu Seiten in der *näheren* oder *älteren Vergangenheit* zurückzukehren.

Aus dem Vergleich der Browsing-Verfahren ergibt sich kein eindeutig bestes Verfahren. Die meisten Verfahren unterstützen effizient das Zurückkehren zu Seiten der näheren Vergangenheit, das Lineare Browsing-Verfahren allerdings nur unter der Voraussetzung, dass die entsprechenden Seiten in der Navigationshistorie nicht verloren gegangen sind. Gegenüber Linearem Browsing haben alle anderen Verfahren den entscheidenden Vorteil, verlustfrei zu sein. Dieser Vorteil kann beim Travel-Browsing nur durch eine zusätzliche Abfrage beim Vorwärts-Navigieren und beim Hyper-Back- und Trace-Back-Browsing durch einen zusätzlichen Button erreicht werden. Verlustloses lineares Navigieren kommt mit zwei Buttons aus, dafür ist das Navigieren zu älteren Seiten nicht effizient. Das Unstetig-lineare Verfahren schließlich verletzt den Erhalt der Nachbarschaftsbeziehung und ist damit in vielen Situationen nicht erwartungskonform.

6 Ausblick

Dieser Beitrag hat aufgezeigt, dass es neben dem weit verbreiteten linearen Browsing-Verfahren weitere Alternativen gibt, welche insbesondere das Problem des Informationsverlusts der Navigationshistorie nicht haben.

Verlustloses lineares Navigieren bietet den Vorteil, dass es sich in die Benutzungsoberfläche bestehender Browser einfach integrieren lässt. Hyper-Back und Trace-Back mit ihrem zusätzlichen Button

scheinen eher für erfahrenere Benutzer geeignet, die die zusätzlichen Möglichkeiten auch zu nutzen wissen. Wir sehen die hier vorgestellten neuen Verfahren weniger als Ersatz der bestehenden Verfahren, vielmehr als zusätzliche Möglichkeit der Individualisierung für den Benutzer.

Alle Verfahren würden durch eine Visualisierung der Navigationshistorie profitieren [11]. Dies gilt insbesondere für das Travel-Browsing-Verfahren, deren Navigationshistorie als einzige eine nicht lineare Grundstruktur aufweist.

Zwischen Browsing- und Undo-Verfahren besteht eine Dualität, auf die wir an dieser Stelle nicht näher eingehen konnten. Unsere Erfahrung zeigt, dass sich Erkenntnisse aus dem einen Bereich in den anderen übertragen lassen. Die Funktionen Zurück und Vorwärts würden dann Undo und Redo [5] entsprechen.

Im aufkommenden Markt der mobilen Kleinstcomputer (PDA, personal digital assistant) kommt es darauf an, den Bildschirmplatz möglichst effizient zu nutzen. Hierfür scheinen Navigations-Verfahren, die auf den einfachen Zurück-/Vorwärts-Buttons beruhen geeigneter zu sein.

Die Browser-Metapher hält schon seit einigen Jahren Einzug in herkömmliche Anwendungen, die das Anzeigen vielfältiger Informationen umfassen (z. B. Hilfesysteme, Finanzsoftware [8]). Auch bei E-Learning und E-Commerce ist es wichtig, dass sich die Benutzer effektiv und effizient auf ihrem eingeschlagenen Weg hin- und hernaavigieren können. Gerade dort können die hier vorgestellten Navigationsverfahren nutzbringend für den Menschen eingesetzt werden.

Literatur

- [1] Bilinski, S.; Bumann, S.: *Lost in History - Entwicklung von Strategien zur Minimierung der Orientierungs- und Navigationsschwierigkeiten in besuchten WWW-Seiten*, Diplomarbeit, Fachbereich Informatik, Universität Hamburg, 1998.
- [2] Conklin, J.: *Hypertext: An introduction and survey*. IEEE Computer Vol. 20, No. 9, Sept. 1987, S. 17-41, 1987.
- [3] Dittrich, G.; Tochtermann, K.: *Browsing in unstrukturierten Hypertextdokumenten*, Universität Dortmund, S. 75-86, In: Hypertext - Hypermedia '91: Tagung der GI, SI und OCG; Graz, 27.-28. Mai 1991; proceedings / H. Maurer (Hrsg.) Berlin [u. a.]: Springer, 1991.
- [4] Furnas, G. W.: *Generalized fisheye views*, Bell Communications Research, In: M. M. Man-
tey and P. Orbeton, editors, Proceedings of the ACM Conference on Human Factors in Computer Systems, SIGCHI Bulletin, pages 16-23, New York, USA, 1986.
- [5] Herczeg, M.: *Software-Ergonomie, Mensch-Computer-Kommunikation*, Oldenbourg; Addison-Wesley, München, S. 188-196, 1994.
- [6] Moulden, A.: *CalHTMLPane v2.021* für Java 2, <http://www.netcomuk.co.uk/offshore/>, Stand: 12/2001.
- [7] Persché, R.: *Immediate Feedback During Online Lectures*, Thesis for Master of Science, Hypermedia Unit, Department of Computer Science, University of Auckland, New Zealand, http://www.iicm.edu/rp_feedback/n60, August 1997.
- [8] Rotard, M.: *Analyse und Neuentwurf einer ergonomischen Benutzungsoberfläche zur Unterstützung des Workflows bei Bilanzierungssystemen*, Diplomarbeit, Fakultät Informatik, Universität Stuttgart, 2001.
- [9] Shneiderman, B.; Kearsley, G.: *Hypertext Hands-On! - An introduction to a new way of organizing and accessing information*, Addison-Wesley Publishing Company, 1989.
- [10] Streit, N. A.: *Die Rolle von mentalen und konzeptuellen Modellen in der Mensch-Computer Interaktion: Konsequenzen für die Software-Ergonomie?* In H. J. Bullinger (Hrsg.), Software-Ergonomie '85. Mensch-Computer-Interaktion, S. 280-292, Teubner Verlag, Stuttgart, 1995.
- [11] Tauscher, L. M.: *Evaluating History Mechanisms: An Empirical Study of Reuse Patterns in World Wide Web Navigation*. Department of Computer Science, Thesis for Master of Science, The University of Calgary, Calgary Alberta, S. 14-32, 1996.

Kontaktadresse

Martin Rotard
Universität Stuttgart
Institut für Visualisierung und Interaktive Systeme
Breitwiesenstraße 20-22
70565 Stuttgart
martin.rotard@informatik.uni-stuttgart.de