

Visualizing Web Sessions:
Improving Web Browser History by a Better
Understanding of Web Page Revisitation and a New
Session- and Task-Based, Visual Web History Approach



Matthias Mayer
2007



University of Hamburg



Universität Hamburg
Department Informatik
University of Hamburg
Department of Informatics

Dissertation zur Erlangung des Grades des Doktors der Naturwissenschaften,
eingereicht am Department Informatik der Universität Hamburg.
*Dissertation in the Department of Informatics submitted to the faculty in partial fulfillment of the
requirements for the degree of »Doctor rerum naturalium« at the University of Hamburg.*

Gutachter | *Reviewers:*

Prof. Dr. Benjamin B. Bederson, University of Maryland, USA
Prof. Dr. Horst Oberquelle, Universität Hamburg, Deutschland
Prof. Dr. Christopher Habel, Universität Hamburg, Deutschland
Tag der Disputation | *day of disputation:* 12. Dezember 2007

Widmung | Dedication

Meiner Mutter Esther,
meinem Vater Karl,
meiner Schwester Gabriele.
In großer Dankbarkeit.

*I am grateful to
my mother, Esther,
my father, Karl,
my sister, Gabriele.*

Zusammenfassung

Bei der Benutzung des World Wide Web spielen verschiedene Arten des Wiederbesuchs von Webseiten (Revisit) eine wesentliche Rolle. Kurzzeit-Revisits entstehen häufig beim Explorieren von Websites und von Suchergebnis-Listen. Mittelfristig entstehen Revisits vorwiegend durch die Wiederbenutzung bevorzugter Werkzeuge und durch das Beobachten von Seiten, deren Inhalt sich häufig ändert. Langzeit-Revisits entstehen vor allem durch das erneute Lesen spezifischen Inhalts nach einiger Zeit. Die heute verfügbaren History-Werkzeuge wie der Zurück-Knopf (Back Button), Lesezeichen (Bookmarks), Verlauf (History List) und die automatische Vervollständigung einer Adresse (Auto-Complete) bilden ein heterogenes Umfeld unterschiedlichster Ansätze, die alle ihre eigenen Methoden des Zugriffs auf die persönliche Web-History bieten und mit den verschiedensten Benutzungsschnittstellen ausgestattet sind. Jedes dieser Werkzeuge besitzt charakteristische, teilweise erhebliche Schwächen. Bestehende Vorschläge zur verbesserten History-Unterstützung schaffen teilweise Abhilfe, bringen aber jeweils neue Probleme mit sich. Dies führt zu der gegenwärtigen Situation, in der eine angemessene, integrative History-Unterstützung noch immer fehlt.

Die vorliegende Arbeit analysiert den Status Quo des Web bezüglich Navigation und Wiederbesuch. Sie vergleicht Vor- und Nachteile gegenwärtiger und vorgeschlagener History-Werkzeuge. Auf Grundlage dieser Erkenntnisse wird ein neuer Ansatz für ein integrierendes Web-History Werkzeug vorgestellt – der Session-Graphs Ansatz. Dabei wird der Pfad des Benutzers als animierter, interaktiver Graph visualisiert, wodurch eine Verarbeitung vorwiegend auf perzeptueller anstatt auf kognitiver Ebene ermöglicht wird. Das Layout der Graphen wird automatisch berechnet, um möglichst charakteristische visuelle Formen zu schaffen, die eine Wiedererkennbarkeit auf Sitzungsebene aber auch auf Ebene einzelner Knoten gewährleisten. Die gesamte History eines Benutzers wird entsprechend einzelner Sitzungen unterteilt, so dass der Umfang einer Visualisierung begrenzt bleibt. Der Benutzer kann das Layout der Graphen manuell nach seiner Vorstellung ändern. Die einzelnen Sitzungs-Visualisierungen können nach übergeordneten Aufgaben organisiert und archiviert werden. Dadurch berücksichtigt das vorgestellte Verfahren im Gegensatz zu anderen existierenden und vorgeschlagenen Ansätzen auf konzeptueller Ebene bereits die Skalierbarkeit hinsichtlich wachsender Histories.

Eine prototypische Realisierung des Ansatzes in Java, basierend auf dem Jazz-Grafik-Toolkit und dem Scone Framework für Web Erweiterungen, wurde entwickelt und in zwei kontrollierten Experimenten und einer Langzeitstudie verwendet, um wesentliche Aspekte des Ansatzes zu evaluieren. Zusätzlich wurden bestehende, teilweise überholte Einsichten bezüglich Navigation und Wiederbesuch im Web zusammen mit drei Kollegen in einer Langzeit-Click-Stream-Studie vertieft und auf den neuesten Stand gebracht. Diese Analysen bilden eine solide Grundlage zur Diskussion und weiteren Verbesserung der vorgestellten Konzepte.

Abstract

When people use the World Wide Web, they show various forms of web page revisitation.¹ In the short-term, web pages are revisited during the exploration of various areas of a website and while following different links of a search results list. In the medium term, revisits occur when people reuse their favorite web-based services or when they observe modifications of well-known frequently updated pages. Long-term revisits mainly occur when specific content is reviewed after considerable time. Current history tools such as the back button, bookmarks, history list, and query autocompletion build a heterogeneous mixture of different approaches, all with different kinds of history access and interfaces. Each tool shows unique and often severe shortcomings. Existing suggestions for improving history support overcome some of these shortcomings, but introduce new ones. An appropriate, integrative history support, therefore, is still missing.

This dissertation analyzes the current situation of navigation and revisitation on the web. It outlines and compares current and suggested history tools, as well as their advantages and disadvantages. Based on these insights, a new visual approach for an integrative history tool is presented – the SessionGraphs approach. It visualizes the user's path as an animated, interactive graph, utilizing perceptual abilities and recognition instead of recall. Session borders are used to subdivide the visual history into items of reasonable, manageable size. A session is laid out automatically with the goal of maximizing the recognizability of the entire session's graph, as well as of the single nodes contained within it. The resulting characteristic visual shapes can be manually reshaped on demand. The user can then organize the visualizations according to the tasks he or she is working on. Thus, this approach provides a conceptual strategy to scale with growing histories – an important issue, which is not explicitly addressed by other existing and suggested approaches.

A prototypical implementation in Java, based on the Jazz Graphics Toolkit and the Scone framework for web enhancements, is used to evaluate the approach in two controlled experiments and a longitudinal case study. Existing and partly outdated insights about navigation and revisitation behavior on the current web without additional history support are updated and deepened in a longitudinal client side clickstream study, executed together with three colleagues. These analyses and evaluations provide a thorough base for the discussion and further improvement of the presented approach.

¹ From now on the term *web* will be used instead of *World Wide Web* and refers to all objects that can be accessed with a common web browser, as described on page 259.

Acknowledgements

The presented work is the result of productive collaborations and the unremitting scholarly and emotional support of great colleagues and friends. My deep gratitude is with all of them.

I thank Peter Schefe for motivating me to start this venture, for his support and guidance along the way. I thank Ben Bederson and Ben Shneiderman for their invitation to an extremely fruitful residence at the Human-Computer Interaction Lab (HCIL) at the University of Maryland, for their excellent support, affectionate reception, and encouragement to continue this work. I would like to thank Ben Bederson in particular, for his knowledgeable, thoughtful advice related to programming, evaluation, and research direction, and his extraordinary hospitality.

I thank Horst Oberquelle and Christopher Habel of the Department of Informatics at the University of Hamburg for their continuous support, including all things organizational, and for supporting me in carrying through. I especially thank my friend and colleague, Harald Weinreich, who read the whole thesis – partly more than once, for his passionate scientific collaboration and precious feedback. My gratitude is also with Hartmut Obendorf and Eelco Herder for their great cooperation. I thank my colleagues Ruth Meyer, Björn Gehlsen, Holger Mügge, and Ralf Bachmann for making work and research a joyful experience. I thank all my former students and participants who contributed to this project, especially Volkert Jürgens, Torsten Hass, Frank Wollenweber, and the highly helpful and skilled people at the HCIL, in particular Jesse Grosjean, Lance Good, Harry Hochheiser, Hilary Browne, Anne Rose, and Catherine Plaisant. I appreciate the contributions of Shaun Kaasten and Gene de Lisa, and the discussions with Yuri Engelhardt. I thank the great librarians of the Department of Informatics, the administrative and computer staff, and the service team.

Last, but not least, I would like to thank my mother, my father, my sister Gabriele Sperling, and my friends for their deep encouragement, love, and patience through all the years that brought me to this point in time. Especially, I thank Susanne Reizlein, Frank Misiak, Jan Heise, Inga Wellmann, Nina Hansen, Luise Palace, the people @ futuretank, and Ursula McKnight for her great corrections. I thank the people at *Haus 73*, *Pony Bar*, and *Unter den Linden* for the wonderful environments they create, their smiles, and their vital soya latte.

Thank you!

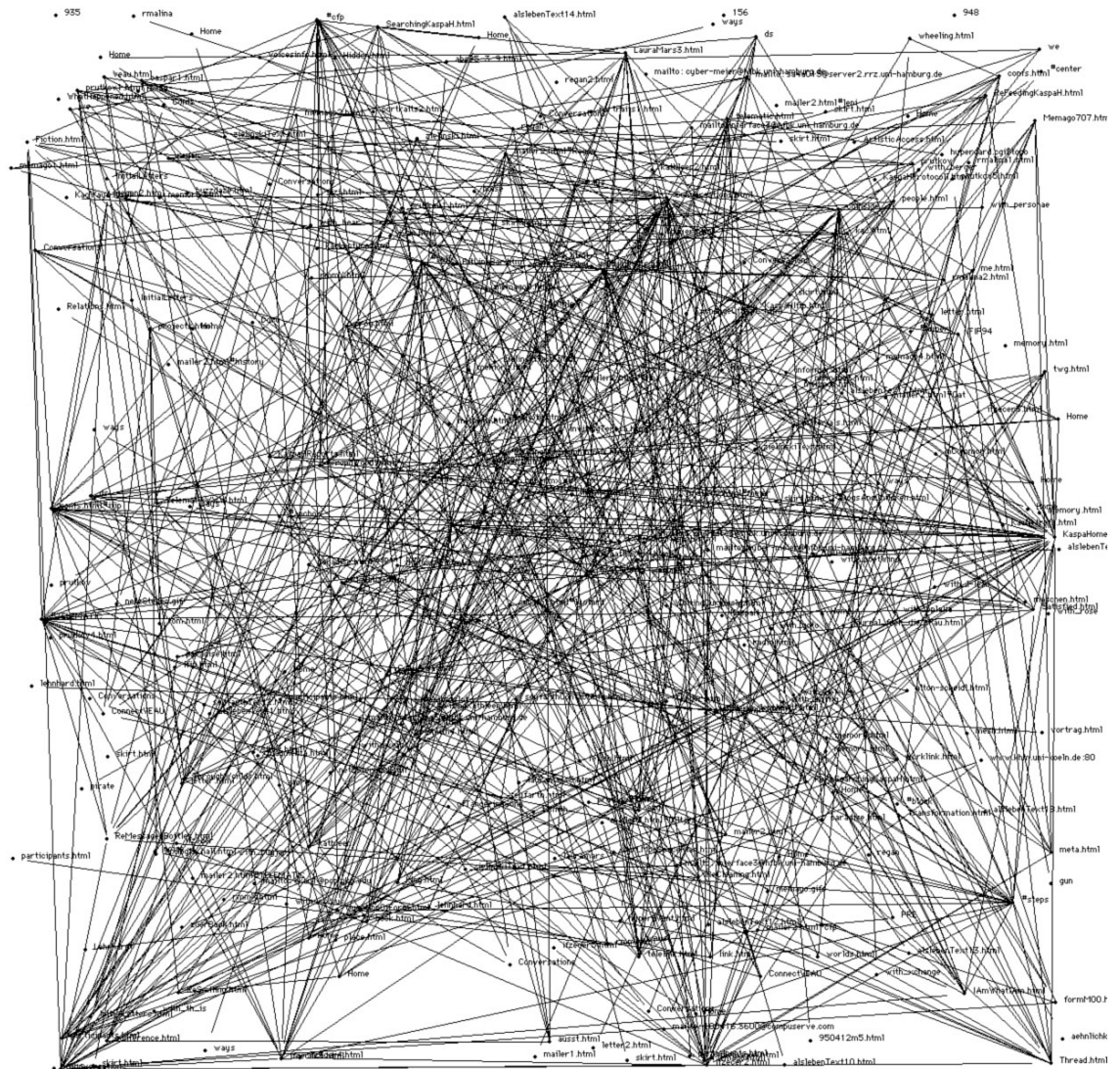


Figure 2-1: "Depicting the Mesh" - web visualization by the author, 1993.

This structure of interlinked web pages emerged during a conversational process amongst the participants of the Interface 3 conference 1993 (Dencker 1997) - I wrote a small program that plotted them randomly on a 2D surface.

Table of Contents

List of Figures	xi
List of Tables	xiii
1 Introduction.....	1
2 Background and Status Quo.....	9
2.1 Revisiting the Web.....	9
2.1.1 The Significance of the Web.....	10
2.2 Revisitations and the Hypertextual Web	13
2.2.1 The Web's Hypertextual Structure.....	13
2.2.2 Navigation on the Web	16
2.2.3 Revisitations, Tasks, and Sessions	19
2.2.3.1 Tasks	19
2.2.3.2 Information Gathering Tasks.....	24
2.2.3.3 Sessions	25
2.2.4 Defining Web Page Revisitation	28
2.2.5 Status Quo of Revisitation Behavior Analysis	32
2.3 Status Quo of History Tools.....	37
2.3.1 History Support in Current Web Browsers	37
2.3.1.1 Backtracking	38
2.3.1.2 Bookmarks.....	41
2.3.1.3 History Lists	43
2.3.1.4 Further History Techniques in Current Browsers.....	45
2.3.2 History Visualization and Related Projects.....	47
2.3.2.1 Applying Information Visualization.....	47
2.3.2.2 The Path as 2D Spanning Tree	49
2.3.2.3 The Path as 2D Graph	52
2.3.2.4 The Path as 1D List	54
2.3.2.5 Further Web Visualizations	56
2.3.2.6 Summary.....	61
2.3.3 Seven Critical Situations of Revisitation	62
3 The SessionGraphs Approach for Revisitation.....	67
3.1 Problem and Goal	67
3.2 The SessionGraphs Concept.....	70
3.2.1 Supporting Sessions and Tasks	70
3.2.2 Applying Visualization.....	72
3.2.2.1 Visualization of Sessions.....	72
3.2.3 Providing an Integrative Interface that Supports Flow.....	82

3.2.4	Scenario of Use – Addressing the Seven Situations	92
3.2.4.1	How the 7 Situations of Revisitation are Supported.....	94
3.3	Prototypical Realization.....	98
3.3.1	Software Architecture.....	100
3.3.1.1	Integrated Frameworks and Libraries.....	102
3.3.1.2	Internal Architecture.....	107
3.3.2	Implementation	110
3.3.2.1	Static Structure – Packages and Important Classes.....	110
3.3.2.2	Dynamic Behavior	114
4	Empirical Evaluation of the SessionGraphs Approach	135
4.1	Methodical Approach.....	135
4.2	Two Controlled Experiments.....	137
4.2.1	Participants, Equipment, and Stimuli	137
4.2.2	Controlled Experiment 1 – Short-Term Revisits	139
4.2.2.1	Method and Procedure	139
4.2.2.2	Results for Short-Term Revisits	142
4.2.3	Controlled Experiment 2 – Long-Term Revisits	143
4.2.3.1	Method	145
4.2.3.2	Results	145
4.2.4	Modifications According to Participants’ Feedback	146
4.2.5	Limitations of the Experiments	147
4.2.6	Conclusion	147
4.3	Longitudinal Study	148
4.3.1	Participants, Prototype, and Equipment.....	149
4.3.2	Method and Procedure – Interviews and Logging.....	150
4.3.3	Results.....	153
4.3.4	Conclusion	167
5	Re-Studying Re-Visitation	171
5.1	Reasons For Re-Studying	171
5.2	Method, Participants, Preprocessing.....	172
5.2.1	Detection of Technical Sessions	174
5.3	Results.....	174
5.3.1	General Measures.....	174
5.3.2	Less Back, More Windows, Much Google	178
5.3.3	Sessions	182
5.3.4	Infrequent Revisits.....	189
5.3.5	Multiple Windows and Tabs.....	190
5.3.6	Revisits to Groups of Pages	192
5.4	Conclusions	199

6	Suggestions for Further Improvement and Future Work.....	203
6.1	Five Specific Suggestions.....	203
6.1.1	Less Cluttered Views: A Denser Layout	204
6.1.2	Reduced Overhead: Automated Session & Task Organization.....	207
6.1.3	Improving Look-Ahead Graphs	214
6.1.4	Visualizing Searches.....	215
6.1.5	Multiple Window and Tab Support.....	216
6.2	Additional Recommendations for Future Work.....	220
6.2.1	Re-Implementation.....	220
6.2.2	Further Evaluation	221
6.2.3	Modifications of the Existing Approach.....	221
6.2.4	Extending the Vision.....	223
7	Conclusion and Contributions	227
8	Used Literature	231
9	Appendices.....	251
9.1	Personal Epilogue	251
9.2	Technological Background.....	255
9.3	Data Related to Prototypical Implementation.....	261
9.3.1	Classes Overview and Source Lines of Code (SLOC).....	261
9.4	Data Related to Two Controlled Experiments.....	263
9.4.1	Consent Form	263
9.4.2	Experimental Tasks and Questionnaire.....	263
9.4.3	Participants' Comments	270
9.4.3.1	First Meeting – Short Term Revisits	270
9.4.3.2	Second Meeting - After Some Days.....	273
9.5	Data Related to Longitudinal SessionGraphs Study.....	276
9.5.1	Online Tutorial and Help.....	276
9.5.2	Tutorial Movies	286
9.5.3	Consent Form	288
9.5.4	Questionnaire for First Interview	288
9.5.5	Questionnaire for Second Interview	290
9.5.6	Logged Event Types	292
9.6	Data Related to Collaborative Web Usage Study.....	293
9.6.1	Individual Contributions of the four Authors	293
9.6.2	SGUSs-Detection-Examples	295
9.7	Formalia	302
9.7.1	Eidesstattliche Erklärung.....	302

List of Figures

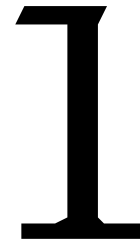
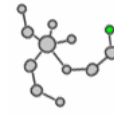
Figure 1-1: “Depicting the Mesh” – web visualization by the author, 1993.....	vii
Figure 2-1: Growth of the internet – active hosts.	11
Figure 2-2: Tasks & hierarchical-sequential structure (cf. Volpert 1994: 20).....	22
Figure 2-3: Example of a structured task.....	23
Figure 2-4: Towards a notion of session – Rosenberg’s episode (1996: 25).	27
Figure 2-5: NCSA XMosaic 2.6 browser with Hotlist View.	33
Figure 2-6: Usage share of web browsers, 1994-2006 (Wikimedia 2007).....	38
Figure 2-7: Microsoft Internet Explorer 6.0 with history & navigation tools.....	38
Figure 2-8: History list of Microsoft Internet Explorer 6.0 (3 views).	44
Figure 2-9: The WebMap history interface.	49
Figure 2-10: The PadPrints interface.....	50
Figure 2-11: The Domain Tree Browser interface.	51
Figure 2-12: The TrailBlazer interface – an advancement of DTB.	52
Figure 2-13: The Nestor Navigator interface – the path as a graph.....	53
Figure 2-14: The WebNet interface – the path on a regular grid.....	54
Figure 2-15: The WebView interface.....	55
Figure 2-16: The GroupLab interface – an advancement of WebView.	55
Figure 2-17: The VISVIP interface – website and user path.....	56
Figure 2-18: The CVP interface.....	57
Figure 2-19: The WebBook interface.....	57
Figure 2-20: The Web Forager interface – organizing several WebBooks.	58
Figure 2-21: The Data Mountain interface.....	59
Figure 2-22: The Footprints interface.....	60
Figure 2-23: Bertel’s sketch, combined paths comparable to Footprints.	61
Figure 3-1: Early sketches of SessionGraphs visualizations.....	74
Figure 3-2: Cultural uses of the visualization of graphs.....	75
Figure 3-3: Animation algorithm derived by water motion observation.	80
Figure 3-4: The SessionGraphs GUI to the right of a common browser.	84
Figure 3-5: SessionGraphs GUI (1) – menu bar and toolbar.....	85
Figure 3-6: SessionGraphs GUI (2) – the task chooser.	86
Figure 3-7: SessionGraphs GUI (3) – the session view.	87
Figure 3-8: Popup-menu for node modification in session- & task-views.	87
Figure 3-9: SessionGraphs GUI (4) – the task view.....	88
Figure 3-10: SessionGraphs GUI (5) – the detail view.....	89
Figure 3-11: The SessionGraphs configuration settings dialog.	91
Figure 3-12: Look-Ahead-Graphs to support same path revisits.	96
Figure 3-13: SessionGraphs development – phases and prototypes.....	99
Figure 3-14: Early SessionGraphs prototype.....	100
Figure 3-15: Environment of prototype and third party software.	103
Figure 3-16: Structural diagram of an exemplary Jazz scenegraph.....	104

Figure 3-17: Thumbnail generating code.....	105
Figure 3-18: Directory and file structure for local thumbnail storage.	106
Figure 3-19: The prototype’s internal architecture.	109
Figure 3-20: Principal fields and methods of <i>WebObject</i> and <i>Visit</i>	113
Figure 3-21: Events after a page visit.	115
Figure 3-22: A <i>SessionView</i> ’s internal Jazz scenegraph structure.....	118
Figure 3-23: Code that renders either the session- or the task view.....	119
Figure 3-24: Force directed placement algorithm for animation.	120
Figure 3-25: Code that triggers the single animation loops.....	122
Figure 3-26: Smooth animation using the force-directed algorithm.....	123
Figure 3-27: Placing a new node in a session graph at P2.....	124
Figure 3-28: Code that renders miniature graphs.	125
Figure 3-29: Code that treats framesets (1) – <i>SGASconePlugin.update()</i>	128
Figure 3-30: Code that treats framesets (2) – The timer	129
Figure 3-31: Excerpt of a file that stores SG-session visualization data.....	131
Figure 3-32: File structure used to store the history information persistently.....	132
Figure 3-33: User definable GUI settings stored in a file.	133
Figure 4-1: Exemplary study task as presented on screen.	138
Figure 4-2: Home page of the National Park Service as used for the study.	139
Figure 4-3: Example of specific park page.	140
Figure 4-4: Typical target page containing the solution to one task.....	141
Figure 4-5: Results of controlled experiments 1 & 2 (diagram).....	144
Figure 4-6: Excerpt of one user’s general log file (global and daily values).	152
Figure 4-7: Excerpt of one user’s detailed log file (single sequential entries).	152
Figure 4-8: Exemplary session visualizations created by the study participants....	160
Figure 4-9: <i>SessionGraphs</i> interaction techniques as applied by the subjects.....	163
Figure 5-1: Web activities as distributed over the day and over weeks.....	177
Figure 5-2: Daily time spent in technical web sessions.	179
Figure 5-3: Classification of the 300 most frequently visited domains.	179
Figure 5-4: Empirical validation of 25.5 minutes session border timeout.	183
Figure 5-5: Detailed visualization of one participant’s log file.	184
Figure 5-6: How visits are distributed over sessions of different duration.	186
Figure 5-7: Comparing revisits to identical groups versus identical sequences.	196
Figure 6-1: Improved node-placement to create a denser, spiral-like layout.	205
Figure 6-2 a–h: Improved versus previous session visualization algorithm.....	208
Figure 6-3 (next double page): Examples of improved algorithm.	209
Figure 6-4: The <i>NoteLens</i> Interface by Benjamin Bederson.	213
Figure 6-5: Suggestion for an improved search visualization.....	215
Figure 6-6: Multiple-windows situation “monoview”.	216
Figure 6-7: Multiple-windows situation “multiview, unrelated”.	217
Figure 6-8: Multiple-windows situation “multiview, related”.	218

Figure 8-1: GPS visualization from the PHYSIS art project – two weeks’ path.....	252
Figure 8-2: GPS visualization from PHYSIS art project – one afternoon.	253
Figure 8-3: “Bungee / Thread of life” (1990).....	254
Figure 8-4: (Web) document models (Schefe 2001a: chapter 1.6).	256
Figure 8-5: A simple HTML document.....	258
Figure 8-6: Exemplary study screen.....	264

List of Tables

Table 2-1: Nine contexts of navigation and orientation.....	18
Table 2-2: Taxonomy of revisit types along four dimensions.....	30
Table 2-3: The most popular web browsers in July 2006.....	37
Table 3-1: Visual encodings used to reduce cognitive load.....	77
Table 3-2: Gestalt laws used to support SessionGraphs visualizations.	78
Table 3-3: Cost hierarchy of history access that considers the users’ needs.....	83
Table 3-4: Menus and menu items, mnemonics (underlined), and shortcuts.	85
Table 3-5: Toolbar items and their functions.	86
Table 3-6: SessionGraphs implementation – packages and selected classes.	112
Table 4-1: Results of controlled experiments 1 & 2 (see Figure 4-5 for diagram)...	144
Table 4-2: Results of longitudinal study (general measures).	153
Table 4-3: Revisits – over time and across sessions.....	155
Table 4-4: Revisits – interdependency of time and session context.	155
Table 4-5: Session analysis – number of distinct pages (=nodes) per session.	156
Table 4-6: Session analysis – number of distinct hosts per session.	157
Table 5-1: Our study’s results compared with the three major earlier ones.....	175
Table 5-2: General measures of web usage study with common browser.	176
Table 5-3: Analysis of visits and revisits to Google pages.....	181
Table 5-4: Character of measured technical sessions (T-sessions).....	185
Table 5-5: Analysis of intra- and intersession revisits based on T-sessions.	187
Table 5-6: Analysis of revisits to entire groups of pages.....	195
Table 5-7: Analysis of revisits to identical sequences of pages.	196
Table 5-8: Analysis of sessions and contained group revisits.....	197
Table 5-9: Mean weekly number of SGUSs under certain constraints.....	197
Table 6-1: Screen space saving character of the new layout algorithm.....	207
Table 8-1: SessionGraphs code – all classes, source lines of code (SLOC).....	262
Table 8-2: Single contributions of the four authors to the web usage study.....	294



Introduction

Presumably man's spirit should be elevated if he can better review his shady past and analyze more completely and objectively his present problems. He has built a civilization so complex that he needs to mechanize his record more fully if he is to push his experiment to its logical conclusion and not merely become bogged down part way there by overtaxing his limited memory. (Bush 1945: 46)

The following situations demonstrate seven severe problems related to revisitation on the web. Similar situations are likely to happen everyday, everywhere around the world. However, current browsers still address them inadequately. This situation adds up to a massive global productivity loss (Cockburn & McKenzie 2001).

Situation 1 – Ariadne cannot go back

Ariadne, an artificial intelligence expert, discovers on her colleague's home page several interesting links. She follows one branch deep into the site's structure and finally reaches an interesting page about mazes. She goes back to the home page and explores further links. When she wants to revisit the maze page again, unexpectedly, she cannot reach it using the back button. She also cannot remember which link on the home page led to that page. Finally, she is able to find the item in the browser's history list. The procedure, however, wasted an unnecessary amount of time and attention. Ariadne was frustrated without need.

[Back to Another Branch or Window Revisit (S1): *short-term, single-page, review same content*]¹

¹ Classification according to the *taxonomy of revisits* chapter 2.2.4 and chapter 2.3.3.

Situation 2 – Bacchus tumbling between his sessions

While looking for a great wine offer on various websites, all explored in different browser tabs¹, Bacchus opens a new browser window to search for a set of nautical charts a friend told him about. After ordering the maps, he wants to return to this great Retsina wine offer that he discovered in one of the tabs and get a box for his wife Ariadne's birthday party. He is, however, not able to rediscover it easily. It is hidden somewhere deep in some tab's history.

[Parallel session revisit (S2): *short-term, single-page, review same content*]

Situation 3 – Chloris has a hard time resuming her work

For her dissertation on floral patterns in art, Chloris is looking for interesting publications in various online libraries. She has to interrupt her work suddenly, when her supervisor calls her. Four days later she finds time to resume her literature search. Laboriously, she has to revisit each promising page again, arranged in a reasonable set of windows. Unnecessarily, she repeats all of the single actions again – no means is provided to re-build the earlier state. Twenty minutes and a lot of attentiveness are wasted until she is ready to resume her work.

[Session-resumption revisit (S3): *long-term, multi-page, review same content*]

Situation 4 – Daedalus suffers this rediscovery

As an architect, Daedalus accepts a bid for a huge, complex building. He remembers an inspiring picture drawn by an infamous artist that he occasionally discovered on the web while exploring several websites about two months ago. Now he sits down for over an hour to retrieve this picture, however, he is not able to locate it again.

[Rediscover revisit (S4): *long-term, single-page, review same content*]

Situation 5 – Echo is repeating her path again and again

Since Echo is overburdened by her crowded bookmark archive, she has accustomed herself to a special behavior. Whenever she wants to revisit certain important pages in the long-term, she first visits a starting page (e.g. a portal or a home page) and follows several links along a familiar path until she finally reaches the target again. On the way she often becomes distracted and spends considerable time trying to find the proper hyperlinks on the intermediary navigation pages. She knows there should be a faster way.

[Same-path revisit (S5): *long-term, single-page, same or modified content or tool reuse*]

¹ Tabs are GUI elements which provide several view panes in a single browser window, currently becoming increasingly available. They resemble traditional card tabs inserted in paper files and can be selected by clicking an associated text label or icon. At each moment, one tab is the frontmost, active tab.

Situation 6 – Phaeton would love to do it again. However, he can't

Phaeton is working as a designing engineer for automobiles. He is asked to manage a project similar to one that he completed half a year ago. For his web-related managing tasks, such as looking up suppliers' sites and communicating with his co-workers on the intranet, he wants to reuse certain pages again that he already used half a year before. Unfortunately, they are not collected anywhere. Bad luck. He has to rediscover page by page.

[**Similar-task revisits (S6):** *long-term, multi-page, same content*]

Situation 7 – Gaea would love to share

As team leader in a hospital, Gaea often encounters problems in sharing important web-related information with her staff. During the last week, she found several interesting pages related to alvus surgery techniques, distributed on various websites. She wants to inform two staff members about it. Cumbersomely, she manages to revisit some of the interesting pages, copies URLs to an email, which she sends her staff. Harder than necessary.

[**Group-exchange revisit (S7):** *long-term, multi-page, same content*]

The described situations could be solved much easier. Similar situations currently absorb an unnecessary amount of time and cognitive capacities. This dissertation contributes to the clarification of the fundamentals on web page revisitation and presents a new visual history approach with the goal of improving such situations in future.

The Importance of Better History Support

The web, today's most important information medium, provides billions of web pages, is used by millions of people, and is highly valued by them (cf. chapter 2.1.1). However, it is lacking drastically in usability. Two of the most severe usability problems found in 1998 were *finding* and *re-finding* of information (GVU 1998c). The situation has changed considerably since then, in terms of content and usage, but much too little in terms of usability (compare Nielsen 2005; Guernsey 2004). Improved revisitation support could still make the daily work experience of millions of web users easier. The above situations could become a thing of the past.

Most of the frustration, the distraction, and the troubles introduced by inappropriate history support are needless. Instead of concentrating on the technology, people should be able to spend their precious resources on the real tasks they are engaged with. This could save valuable time, free cognitive capacities, and increase the subjective satisfaction – on a global dimension.

Revisitations are crucial in almost every web-based activity (Tauscher & Greenberg 1997a). They vary in character and motivation – they appear in the short and in the long-term, they refer to single or multiple pages, they are motivated by reviewing same or modified content. Our study, presented in chapter 5 (see also Obendorf, Weinreich, Mayer & Herder 2007), showed that people frequently create revisitations in the short-term, when they *backtrack* during the exploration of web-sites or *undo* their activities during the application of web-based tools. In the long-term, they may more or less regularly *re-utilize* favorite web-based services or *observe* modifications to pages of special interest.¹ They may also need to rediscover one or more pages that contain content that is required again.

It was necessary to provide new, promising ideas for supporting the different situations of revisitation on the web more adequately, and to update the outdated insights on revisitation behavior that are based on studies from seven to thirteen years ago – too old to reflect the current status quo (compare chapter 2.2.5). This dissertation contributes to both of these issues.

A New Visual Approach

The history approach presented here, SessionGraphs, supports all kinds of revisits using one consistent, integrative interface. It is provided as a vertical frame next to the web browser, comprising various interactive views. History information is structured and subdivided based on the user's *sessions* and *tasks*. Examples include Bacchus' quest for the box of wine or Chloris' attempt to find interesting literature. Sessions are characterized by a coherent underlying activity. Tasks in the SessionGraphs approach are higher level constructs that may comprise several sessions and sub-tasks. For example, the new project Phaeton was asked to work on could be modeled as such a task that comprises all the sessions necessary to finish this project. Tasks in the SessionGraphs approach allow the organization of all sessions hierarchically.

The core of the approach is the graph-based, dynamically animated visualization of the sessions that a user executes. Whenever the web is used, a graph-based figure of the currently navigated path emerges beside the browser – similar to figures of stellar constellations (see page 74, Figure 3-1, and page 84, Figure 3-4). These figures grow with every new web page that is visited. Any web page – represented by the graphs' nodes – can be directly revisited using the interactive visualizations. By using characteristic spatial layouts and visual landmarks they are easy to distinguish and fast to recognize.

¹ A more detailed distinction of short-term, medium-term, and long-term revisitations is presented in chapter 2.2.4. Long-term revisits are further detailed to very long-term revisits. To refer to the broad temporal spectrum of revisits in general the text often mentions only short- and long-term revisits.

The basic idea of the approach is to provide the user with that history information, at each point in time that is likely to help him at his or her current activity, based on a well-designed cost hierarchy that controls the access to the different parts of the history (see page 82).

The same visualizations are designed to be used in the short-term and in the long-term. In the short-term they can substitute for backtracking. Ariadne could easily go back to the maze page again. Different sessions can be managed concurrently, allowing users like Bacchus to smoothly switch between concurrent sessions. In the long-term, they support the revisitation of single or multiple pages, the resumption of entire sessions, and the execution of similar tasks. Phaeton, for example, could have simply copied all the relevant sessions and tasks he created during his last project to a new task for the new project. He would then have all the references immediately at hand.

The designed visual history should be a tool that is easy and fun to use, serving the user's history-related needs and giving freedom to care about what really matters. The conceptual and prototypical design of this approach builds the core of this thesis. An extensive long-term clickstream study of current navigation and revisitation behavior on the web completes it.

Overview over the Chapters

Chapter 2 provides the *relevant background* for the development and discussion of the SessionGraphs approach and for the web usage study. It reflects on the web's hypertextual properties and their effects on revisitation behavior. A taxonomy of revisits contributes to more clarity in the ongoing discussion. Insights from the three most important web usage studies are summarized and the history techniques of current browsers and of suggested alternatives are presented.

Chapter 3 introduces the conceptual design and technical realization of the *SessionGraphs approach*. It describes the design of the visualization and the algorithms that animate and layout the views. It further presents the Java-based prototypical implementation that was used in the subsequent studies for evaluation.

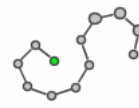
Chapter 4 presents the *empirical evaluation* of the approach. Two controlled experiments compare the approach to a plain Netscape Navigator both in short and long-term revisitations. A longitudinal case study analyzes the acceptance under more naturalistic circumstances and collects participants' feedback on the benefits and possible improvements of the approach.

Chapter 5 introduces *new findings about current revisitation behavior* with unassisted web browsers. In an extensive, long-term client-side clickstream study detailed insights about the character and motivation of current revisitation activities are gained, updating earlier studies from up to 13 years ago. These findings are used to discuss the SessionGraphs approach and to suggest specific adjustments.

Chapter 6 presents concrete *suggestions for improvements* to the SessionGraphs approach and introduces possible *directions for future work*.

Chapter 7, finally, gives a *conclusion* about the presented work and revisits the *single contributions* made in this thesis.

An *epilogue* casts a more personal light on this endeavor and its process. The *appendices* present details about the implementation and evaluation, including an overview of Java classes, technological background, as well as data related to all three studies, such as questionnaire preprints and subjects' comments.



2

Background and Status Quo

The difficulty seems to be [...] that publication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships.

(Bush 1945: 37)

This chapter explores the background of web usage and revisitation. Revisits are considered in their dependency from the hypertextual structure of the web and from the user's tasks that he or she is trying to accomplish when using the web. A brief taxonomy of revisits is provided for more explicit discussion. Current empirical knowledge about revisitation behavior is presented and an overview is given on the history techniques used in today's browsers. In addition, more advanced history visualization projects are introduced. The strengths and weaknesses of these approaches, as well as the concepts of sessions, tasks, and history visualization build the foundation for the SessionGraphs approach, as presented in chapter 3.

2.1 Revisiting the Web

Information has become an essential value of the system we live in (Marchionini 1995). Being informed and being able to find information quickly has become a

central requirement for professional success (*ibid.*). The term *information society* expresses the importance of publicly available information as of 1977 (Porat 1977).

Ideally, any human information need would be satisfied immediately, easily and reliably. However, we are far from achieving this goal. Today people increasingly use the web to satisfy their information needs. As a *tertiary medium* (Pross 1972), the web requires technical appliances both for the production and the reception of its content. These appliances are a barrier between people and information. And instead of decreasing, the barrier seems to grow: “[...] more people must regularly manage more information in order to survive and prosper and they must use an expanding array of technologies to do so” (Marchionini citing Christopher May, 1995: 6). As designers of these appliances we have an immense responsibility to supply people with technology they are willing and able to use; technology which serves their needs, which subordinates itself to their actual tasks. Unfortunately, today’s technology often determines its users’ lives, rather than the other way around. This chapter analyzes the current situation, including the significance of the web, its hypertextual structure, and its users’ tasks. It will become clear, why proper revisitation support is important and why the current situation is far from being ideal.

2.1.1 The Significance of the Web

During the past decade the web has taken the role of the most important medium both for information *dissemination* (Scheffe 2001b: chapter 1.5) and for information *gathering* (USC 2004: 50). It outstripped all competing systems like *Hyper-G* (later *HyperWave*, Maurer 1996) and open hypermedia systems such as *KHS* (Hammwöhner 1997) and the *Devise Hypermedia System* (DHS, Grønbæk, Bouvin & Sloth 1997; Grønbæk & Trigg 1999).¹ Its success can largely be attributed to the conceptual simplicity (Hammwöhner 1997: 22). The high significance of the web today can be ascribed to four qualities as elaborated below – to its large *size*, its widespread *usage*, the broad *variety of uses*, and the good *quality* of information it provides.

The web is large and growing. Since summer 2000, the number of websites that actually provide distinct content more than quadrupled from slightly below 10 million to over 50 million in April 2007. At the same time, circa 114 million different

¹ Further historical information about hypertext systems can be found in (Nielsen 1990b, 1995b; Müller-Prove 2002; Shneiderman & Kearsley 1989). Details are presented by Gilles and Cailliau (2000), Berners-Lee and Fischetti (1999), and the W3C (2000). Individual contributions to the development of the web are listed by Zakon (2006). Earlier hypertext systems are discussed by Shneiderman and Kearsley (1989), Nielsen (1995b), and Müller-Prove (2002).

domain names were registered¹, continuing a strong growth trend since 2005 (compare Figure 2-1). Google (2007) reports that they scan more than eight billion different web pages, which is only a small part of the entire web. The so-called *deep web* – the universe of online documents that are not indexed by search engines – is much larger and contains high quality information (Bergman 2001). It consists of dynamically created documents, e.g. of search results or pages generated subsequent to form submission. Rough estimations assume the *deep web* (or *invisible web*) to be about 500 times as large as the *surface web* (or *visible web*) that is “visible” to search engines (Bergman 2001; Kabra, Li & Chang 2005).

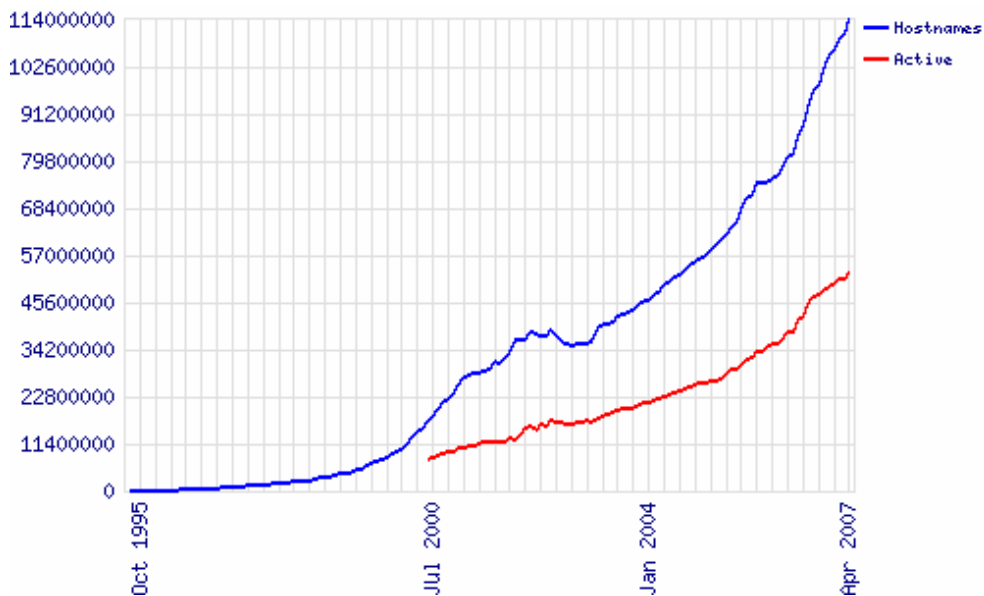


Figure 2-1: Growth of the internet - active hosts.

The lower red line 'Active' represents the number of real websites that offer distinct content. For example news.netcraft.com and news.netcraft.net are two hostnames, but one active site. Jan. 1995 - April 2007 across all domains measured by Netcraft LTD (2007).

The web is massively, globally used. In January 2007 around 747 million people worldwide were using the web, showing an increase of 10% within 12 months (ComScore 2007). In early 2007, more than three quarters (77.6%) of all U.S. Americans age 12 and older were using the internet (USC 2007)² – 68.1% used it at home for almost nine hours per week (ibid.). A study of Europeans in 2006 showed that their mean weekly time spent on the web had doubled in only three years to four hours per week – surpassing the time spent reading newspapers (Jupiter_Research 2006). Experienced users in the U.S. spent already in 2004 even an average of

¹ A number found by counting valid answers to HTTP requests.

² Digital Future Project at University of California (UCLA) and University of Southern California (USC). Studies of more than 2,000 households across the United States.

about 17 hours online per week (USC 2004: 28). Google allows search results to be formatted in 35 different languages and provides its user interface in more than 100 different languages globally (Google 2007).

It is used for a broad variety of purposes. According to the USC survey (USC 2004: 29) the most popular internet activities amongst US citizens are E-mail and instant messaging (90.4%), web surfing or browsing (77.2%), reading the news (52.0%), gathering information about hobbies (46.7%), entertainment (45.6%), and shopping (44.2%). During the past decade, the web has undergone a change from being a plain information source to providing a broad variety of additional services, such as entertainment, shopping, and social platforms. Having once been a static structure of enduring web pages it has grown to a dynamic, fast, and interactive medium coined by web applications and developments such as AJAX¹ and Web 2.0². The web's paradigm has shifted from *publishing* to *participation* (O'Reilly 2005) and "we are now witnessing the true emergence of the Internet as the powerful personal and social phenomenon we knew it would become" (Jeffrey I. Cole, director of the USC Annenberg School Center for the Digital Future – USC 2007). The head of Google Northern Europe, Schindler (2006), forecasts that the web will become a fundamental ingredient of our lives and that it will be the place where buying decisions are prepared and completed, where branding takes place, and where media will be consumed.

Information on the web is highly valued. A last ingredient regarding the web's importance is the high confidence with which web users value the provided information. Almost 70 percent of US users who have access to the Internet at work say that by going online their productivity has improved "somewhat" or "a lot" (USC 2007). 55.2 percent of US citizens consider the web to be a "very important" or "extremely important" source of information (USC 2004: 49). Among the very experienced it even outranks all other media as a "very important" or "extremely important" source of information (USC 2004: 50). Since more and more people worldwide are becoming versed with the web, it may be rightfully assumed that this trend will continue.

Based on the abovementioned evidence for the web's enormous size, influence, and acceptance, it should be a primary goal to provide the user with adequate

¹ AJAX (Asynchronous JavaScript and XML) is a combination of web development techniques used to create interactive web applications. Small amounts of data are transmitted from time to time so that not the entire web page has to be reloaded whenever a change is requested. AJAX makes use of XHTML or HTML, CSS, DOM and ECMAScript.

² Web 2.0, a term coined by O'Reilly Media in 2004, denominates the turning point of the web after the bursting of the dot-com bubble in 2001. It refers to more matured web applications and sites, especially with aspects of participation and collaboration (see also O'Reilly 2005)

usability. Accomplishing this goal, however, still seems to be far away. Nielsen (2001) reports that “User success rates on e-commerce sites are only 56%, and most sites comply with only a third of documented usability guidelines”. This number is startlingly low considering the high interest that providers of e-commerce websites should have in their visitors’ success. Good revisitation support plays a crucial role when striving for proper usability. Once a piece of information has been found for the first time, it should be much easier to find it again, whenever needed (Jones, Dumais & Bruce 2002). Providing adequate history support could even, to some degree, compensate for the lacking usability of a website’s navigational means, since navigation of the site could be partly substituted by using the history tool.

2.2 Revisitations and the Hypertextual Web

Revisitations on the web occur mainly for two reasons. The first one is due to the web’s *hypertextual structure*. Making use of a hypertext means to navigate its structure. This navigation, however, is essentially based on revisitations, especially in the short-term. The second reason is the *user’s task structure*. Repetition and resumption are important parts of human task execution. On the web, these activities predominantly result in long-term revisitations. Consequently, to understand why the usage of the web brings along revisitations it is necessary to investigate both its hypertextual structure and the user’s task structure more closely.

2.2.1 The Web's Hypertextual Structure

Initially, when founded by Tim Berners Lee at CERN¹, the web was intended to be a medium for presenting and exchanging *scientific information* mainly within the research community (Berners-Lee 1989, 1996, 2003; Berners-Lee & Fischetti 1999). It was designed to be rather simple in its structure and to incorporate only selected aspects of the early visions of hypertext, as largely contributed by the so-called hypertext trinity – Vannevar Bush with his idea of the Memex (Bush 1945), Doug Engelbart with his Augment/NLS system (Engelbart 1963, 1962; Engelbart & English 1968), and Ted Nelson with his Xanadu concept (Nelson 2006). Consequently, the first websites provided a simple hypertext based on static text, hyperlinks, and images.

Soon, however, the web left its intended track and started to disseminate a broad variety of content and topics. The need for multimedia elements such as ani-

¹ Conseil Européenne pour la Recherche Nucleaire.

mation, film and sound was successfully met and the web's interactivity and dynamism drastically increased.¹ Today we find a highly dynamic web that creates a considerable amount of content just on demand: Entire applications, such as image editors, email clients, spread sheets, and word processors can be found on the web and have little to do with hypertext. Most websites, however, can still be characterized as hypertexts.²

Structurally, the web is a “pool of (partially) interconnected webs” (Baeza-Yates & Ribeiro-Neto 1999: 68) – a pool of self-maintained websites that define links to other websites and that provide their own interlinked content. A site's content may comprise hundreds or thousands of single web pages, independent from all other sites. As *local hypertext*, a single website is capable of presenting consistent internal structures and unified concepts for *within-site-navigation*, *interaction*, and *visual identity*. This consistency, however, is not given for the navigation between sites. Using the web means to navigate this “web of webs”, both within and between the sites.

Each website that provides static or dynamic web documents can be considered a hypertext itself, as can be the entirety of all websites on a higher level. Being a hypertext means being *non-sequential*: “Hypertext is *non-sequential*; there is no single order that determines the sequence in which the text is to be read” (Nielsen 1995b: 1, emphasis in original). As a *link-based* hypertext³, the web's structure is defined by an underlying directed graph of *nodes*⁴ and *hyperlinks*. Single web pages are connected by directed links⁵, determined by start and end anchors, defining both the structure within and between sites. A link on the web is embedded in the document where its start anchor is located. Mostly, this start anchor acts both as information in itself (either as text or graphics) and as an interactive link anchor. Activating this anchor results in traversing the link to its destination node, a web page that usually replaces the departure page in the browser. *Associative links* are directly integrated into the

¹ The terms *hypertext* and *hypermedia* will be used interchangeably in this text. Some authors retain the term *hypermedia* for hypertext that includes various media beyond plain text (images, sound, movies, 3D data, etcetera).

² The term *website* is used to refer to a sub-set of the web defined by a common domain name. It mostly provides topically coherent content and is maintained by one institution or individual.

³ Other forms of hypertext comprise *spatial hypertext* (Marshall, Shipman & Coombs 1994), *relation based hypertext* (Aquanet - Marshall, Halasz, Rogers & Janssen 1991; Intergrams - Rosenberg 1993), and *set based hypertext* as VIKI (Marshall et al. 1994) and HyperSet (Parunak 1991).

⁴ While the term *node* refers to the *syntactical* entity that defines the text's structure, the semantic term *lexia*, introduced by Roland Barthes (Barthes 1990), refers to units of textual *meaning* (Landow 1992). On the web nodes and lexia are often congruent.

⁵ The terms *link* and *hyperlink* will be used interchangeably.

content of a page and reference related information, whereas *structural links*, often provided in marginal areas, enable the user to navigate the website's structure (on the discussion of link qualities see Nielsen 2000; Weinreich, Obendorf & Lamersdorf 2001; Bieber, Vitali, Ashman, Balasubramaniam & Oinas-Kukkonen 1997; Berners-Lee 1997; Andrews & Dieberger 1996). The navigation along these links is likely to create many revisits in the short-term, as further elaborated below. More details about the technological, hypertextual background of the web are outlined in appendix 9.2, comprising topics such as HTML, HTTP, URIs, XML, and the XML Linking Language.

Most websites are descriptive, hierarchical hypertexts. Compared to sophisticated models of hypertext, such as the Dexter reference model (Halasz & Schwartz 1990, appendix 9.2) the web exhibits a rather simple form. Most content is targeted to satisfy needs for *information* or *service* as fast and as well as possible (Landow 1994b; Landow 1997). This is understandable considering that most web pages (83%) are commercially motivated (Lawrence & Giles 1999) and authors strive for reputation or successful business. Thus, the web contains mostly *descriptive hypertext* in contrast to *narrative* or *argumentative hypertext* (Scheffe 2001a). Narrative hypertexts exploit means of narration to a much greater degree – e.g. they may use disorientation as an intended means (see Joyce 1995, 1992; Keep, McLaughlin & Parmar 1993; Moulthrop 2006). Documents on the web, however, are written in a concise and factual style containing clear, easily conceivable and “scannable” statements (Lynch & Horton 2002a; Morkes & Nielsen 1997; Nielsen 1997b). While web content used to be created by a few authors and read by many readers¹, current participative trends (Web 2.0: blogs, forums, wiki webs, etc.) increase the tendency to intermingle readers and writers – as already introduced by Landow's concept of the *wreader* (1994a: 9 and 14). However, while web users increasingly participate in content creation, navigational structures are still mostly determined by the offering authorities.

The web's structure, similar to its content, mainly follows the needs of an information environment that is easy to understand and easy to navigate. Most websites are hierarchies of little depth, moving from a general overview, provided on the home page, down through increasingly specific submenus and content pages. They follow a design style as suggested by Lynch et al. (2002b). They recommend a hierarchical structure with a width of five to seven and a depth of one to three. Often, the sites' hierarchies are based on metaphors such as libraries, encyclopedias, or organizations. Advocates of “real” hypertext, like Ted Nelson (2003), tend to express their indignation about how far away from many precious concepts of hypertext the current web is: “The web isn't hypertext, it's DECORATED DIRECTORIES!”. These

¹ The terms *reader* and *user* (as well as *writer* and *author*) are used interchangeably.

structural conditions have to be considered when navigation and revisitation support is designed.

2.2.2 Navigation on the Web

The web user is offered two main techniques for navigating the web's structures – *link following* and *querying* (see Marchionini 1995: 8; Liebscher & Marchionini 1988; Marchionini & Shneiderman 1988). Both navigation techniques are often applied during the same activity (Belkin, Marchetti & Cool 1993).¹

Link Following and Querying

Link following denominates the classical way of navigating a link-based hypertext – navigation along connected nodes by traversing hyperlinks. Proper navigation support is essential, since for the reader it is a much higher cognitive challenge to navigate a non-sequential hypertext structure than to move on to the next element in sequential media. Link following often results in many short-term revisits, mostly created by backtracking during the tentative back and forth movements in hub and spoke navigation patterns when a user compares several items, explores a certain area, or wants to reach a desired target. Long-term revisits occur in combination with link-following when the user reuses a familiar path to frequently visited pages.

Querying describes the process of navigating a hypertext by using a search engine and subsequently visiting selected search results – be it a web wide search engine such as Google or a site-specific local engine. Its success depends on the scope of the underlying database and on the ability to find relevant items by increasing recall and / or precision.² Querying allows the entire represented domain to be searched, not only directly interlinked pages. It is characterized by formulating the query, scanning the results, reformulating the query if necessary, and exploring the results. Revisits occur in the short-term when users revisit search results lists or query pages to modify a request. In the long-term, search engines and search results lists are revisited both for the purpose of reuse, as well as to rediscover items of special interest.

¹ Navigation in this context comprises all activities that lead to the traversal of web based lexia. It includes both the *cognitive activities* of the decision making process concerning which direction should be chosen next, as well as the *physical activities*, UI interactions that cause the web browser to execute the intended traversal.

² A document is relevant if it is judged 'useful' in the context of the query, depending on the seeker and the seeking task (Cooper 1971; Wilson 1973). The *recall* weighs the number of retrieved relevant documents against all relevant documents in the database (if the database contains 1000 relevant documents and 100 are retrieved, the recall is 10%). *Precision* compares the number of relevant and retrieved documents to the total number of retrieved documents (if 100 documents are retrieved and 5 of them are relevant, the precision is 5%; the residue of 95% irrelevant documents is called *fallout*).

Nielsen reports that the majority of users prefers querying over link following (Nielsen 1997a).¹ Spool et al. (2001a), however, found that this preference depends to a large degree on the design of the specific sites and the navigational means they provide. They showed that users are more likely to continue a site's exploration after link following than after querying (2001b). Thus, they recommended that the providers of commercial sites should encourage link following rather than querying so that the visitors can discover additional content or products along the way.

Both navigation strategies should be supported by web history systems, be it during activities where the users knows exactly where they are and want only to return in order to explore additional directions, or during phases in which orientation is lost. In this latter case, history systems become even more important.

Orientation and Disorientation

The web requires excellent orientation support. As Bernstein (1988: 43f) already underlines "The promise of hypertext can only be realized if hypertext reading is both pleasant and efficient. Disorientation is unpleasant and wastes time; it must be avoided wherever possible". He addresses one of the biggest problems of hypertext usability – the problem of *getting lost in hyperspace* (see also Conklin 1987; Nielsen 1995b: 247; Edwards & Hardman 1989; Xu, Cockburn & McKenzie 2001). It is likely to happen during link following, but may also appear during querying, e.g. when a user loses orientation due to a confusing number of search results pages, all opened in different browser tabs.

A user can be considered *lost* or *disoriented* when he or she is no longer able to answer the following three questions (Engel, Andriessen & Schmitz 1983; Fitter 1979; Nievergelt & Weydert 1980; Nielsen 2000: 188; Nievergelt 1983): Where am I? Where have I been? Where can I go?

Important instruments to fight disorientation are adequate orientation support and good history systems. The latter primarily help in answering the question "Where have I been?". Within the scope of the user's history – all pages he visited before – they also help answering the question where the user currently is, and where he or she can go from there.

In order to enable a more detailed discussion, I introduced a nomenclature of nine largely independent contexts of navigation and orientation (Mayer 2000), as summarized in Table 2-1. They are categorized into three groups based on the structure, content, and usage of the environment in which a user navigates. Each context refers to a subset of web pages that it comprises. For example, when the user is able to answer the question "where can I go from here?" based on helpful information

¹ "Usability studies show, that more than half of all users are search-dominant, about a fifth of the users are link-dominant, and the rest exhibit mixed behavior. The search-dominant users will usually go straight for the search button when they enter a web site" (Nielsen 1997a).

given in link anchors, he or she is oriented in the “*outgoing*” *link context* – all the pages that are accessible from the current document by traversing one of its links. Correspondingly, orientation in the *context of similarity* is given by answering questions such as “where do I find a similar document?”, and could be provided by a search engine that allows to process similarity-based queries (compare Google’s function “show similar pages”). Orientation in the *context of history* is given once the user is able to answer the question “Where have I been before?”. Orientation in the *session context*, a subset of the history context, is given once the user can perceive all the items that were visited during one session.

Moreover, a user can be properly oriented in one context, yet completely dis-oriented in another one. The goal should be to provide the user with the best possible orientation and means of navigation in all contexts in which the authors want their readers to have orientation and to be able to navigate.

Structure	<ol style="list-style-type: none"> 1. Link Context (defined by associative links, in- and outgoing) 2. Context of Composition (defined by hierarchy of elements) 3. Guided-Tour-Context (author-defined sequential structure)
Content	<ol style="list-style-type: none"> 4. Context of Similarity (defined by specific attributes of the content). 5. Context of Topic (defined by the content’s topics)
Usage	<ol style="list-style-type: none"> 6. History Context (comprising all visits of one user) 7. Session Context (comprising visits of one session) 8. Social Context (comprising all histories of an entire work group) 9. Task Context (defined by the user’s tasks)

Table 2-1: Nine contexts of navigation and orientation.
According to (Mayer 2000). Bold contexts are addressed by SessionGraphs.

For contexts 1 to 5 this can be accomplished by the authors themselves since only within-site information is involved. The support of navigation and orientation within the *contexts of usage* (6 to 9), however, requires adequate browser based history tools since these contexts comprise site-spanning activities. The SessionGraphs approach covers three of them. It allows the user to answer questions such as “where have I been before?” (*history context*, by keeping references to all earlier visited pages), “where have I been during the current session?” (*session context*, by separating single sessions) and “where have I been while working on the task XY?” (*task context*, by providing the ability to organize the history according to the user’s tasks).

2.2.3 Revisitations, Tasks, and Sessions

Besides the aforementioned causes for revisitations, all based on the hypertextual structure of the web, the second important cause for revisitations refers to the user's *tasks* and their partially recurrent structure. This section introduces a general model of *task execution* as it is underlying the developed approach, it describes *information gathering tasks* on the web in more detail, and establishes the notion of a *session* to describe a reasonably small unit of subjectively coherent activities on the web.

People use the web for a broad variety of tasks (see USC 2004: 29 and our study, chapter 5). Many of them relate to some kind of *information gathering* and are embedded in superior processes, such as decision-making, problem solving, learning, being creative, or sense-making (Russell, Stefik, Pirolli & Card 1993; Dervin 1983). Executing such tasks leads to single activities like searching and comparing information about books, publications, news, offers, institutions, persons, products, songs, technologies, etc. In contrast to other web-based activities, such as web-mailing, music-listening, or chatting – information gathering often requires the user to navigate across the borders of single sites. Within-site navigation support in these instances is currently not sufficient and future browser based history tools should provide adequate assistance. In the SessionGraphs approach, this will be addressed by the concepts of sessions and tasks. The task concept is based on a theory about the structure of human activities in general, as presented next.

2.2.3.1 Tasks

During the last 40 years, human work has been thoroughly studied in fields as diverse as organizational behavior, management science, or cooperative work. Information work, as conducted by managers, researchers, financial analysts, consultants, accountants, or administrators is characterized by highly demanding workloads, multiple activities, and the involvement in various projects, initiatives and teams (González & Mark 2004). In academia, for example, a person might supervise students, perform departmental duties, and participate in different research projects. Most studies focus on managers, especially before IT was commonplace (Hudson, Christensen, Kellogg & Erickson 2002; Sproull 1984; Mintzberg 1973; Horne & Lupton 1965); a review is provided by (Panko 1992). It has been shown that the activities of CEOs, for example, can be characterized by their brevity, variety and fragmentation (Mintzberg 1973, 1970).

González and Mark (2004) revealed interesting insights about the activities of current information workers. They observed fourteen people – business analysts, managers, and developers – during their daily work. They organized their work in terms of large, “thematically connected units of work” (ibid.: 113) that were, however, highly fragmented. Interruptions and the frequent switching between different activities, however, can severely tax an individual. González and Mark differentiate

between low level “events”, i.e. “any continuous use of a device or engagement in an interaction with other individuals”, (ibid.: 115); on average, their subjects spent only about three minutes on one event before switching to another one; and, higher level “working spheres”, defined “as a set of interrelated events, which share a common motive (or goal)” (ibid.: 117), i.e. as “higher levels of units of work or activities that people divide their work into on a daily basis”(ibid.). But even on this level, their subjects only spent an average of about 12 minutes in one working sphere before switching to another. The observed individuals worked in a mean of ten different working spheres during the observation period of three and a half days. While interruptions can both be beneficial as well as disruptive (Hudson et al. 2002; see also Gillie & Broadbent 1989), González and Mark (2004) came to the conclusion that switching between events within the same working sphere was much less disruptive than switching between different working spheres.

González and Mark criticized the current designs of information technology that “are based on the notion of supporting distinct tasks such as document production, email usage, and voice communication”, but do not allow to be integrated in a meaningful way (2004: 113). They call for better tools which allow “people to integrate their information into cohesive task structures that make sense to them” (ibid.) – “mechanisms that save the state of the information device particular to that working sphere, making it easier to resume work” (ibid.: 119). SessionGraphs is precisely such a tool.

In the SessionGraphs approach the notions – and artifacts – of *sessions* and *tasks* are used to support people in their activities on two separate levels, similar to the introduced distinction of events and working spheres. The higher level notion of tasks will be used to organize the user’s activities into meaningful, hierarchically structured parts, based on the assumption that it is easier for users to make use of the collected history information once it explicitly reflects and externalizes their actual tasks – easier compared to current approaches that only allow navigation through history information based on rather technical, data-driven categories such as date, frequency of visits, or domain name. Thus, a theoretical framework was required in order to model such tasks. A simple and convincing approach is offered by Action Regulation Theory (ART, “Handlungsregulationstheorie”), a development of industrial psychology (Hacker 1986), the relevant aspects of which will be briefly described below.

ART sheds light on the structure of human task execution in general, and is based on Leontjew’s activity theory (Leontjew 1974) and on the work of Miller, Galanter, and Pribram (1973). As a human centered approach, ART assumes an actor with an individual urge for personal growth who permanently wants to discover and encounter new areas of knowledge and skills. Action Regulation Theory has become a stable basis of industrial psychology and is widely used for task analysis (TBS - Hacker & Richter 1980; VERA - Volpert, Oesterreich, Gablenz-Kolakovic, Krogoll &

Resch 1983; VERA - Oesterreich & Volpert 1991; STA - Ulich 1983; KABA - Dunckel, Volpert, Zölch, Kreutner, Pleiss & Hennes 1993). It will be not be used for analysis in the SessionGraphs approach, but for task modeling.

At its core, ART is based on the model for *hierarchical-sequential organization of action* (HSO, “hierarchisch-sequentielle Handlungsorganisation”), which itself is based on two principles (Volpert 1994: 20). First, nested tasks create a logical hierarchy of tasks and sub-tasks. Second, the deepest sub-tasks in the hierarchy define the temporal sequence of activities during task execution.

The execution of tasks, according to ART, is motivated by the perception of a gap between the status quo and a desired goal state. The actor develops and follows a plan to execute the task and reach the goal. Ideally, the process begins with the formulation of the goal and ends with its achievement (Volpert 1994: 18). When working on one task comes to its end, the next task is usually already planned, forming a cyclic process.

The hierarchy in HSO is defined by goals and sub-goals that can be identified with related tasks and sub-tasks. Accordingly, each task is part of a superior task, which itself acts as a sub-task of another superior task (see Figure 2-2). Sub-tasks are assumed to be created partly in advance of all actions, but can also be developed during the process of refining a superior task.

The deepest level of the resulting task hierarchy is determined by basic, learned sequences of activities (*operations*), such as finding a publication in a well-known online-library or ordering an online-ticket on a frequently used website. The upper end of the HSO hierarchy is constrained by general tasks. According to ART, the actor ensures with incidental compatibility checks that the superior tasks are still focused while executing sub-tasks. If a deviation is perceived, an action is modified following the principle of *local modification*: The actor tries to find alternative actions to achieve the task’s goal and to constrain the modifications to the smallest possible local area.

Both principles of HSO – the hierarchy of tasks and the sequence of operations – are mirrored in the SessionGraphs approach; the first one by the user-definable hierarchy of tasks, the second one by the concept of single sessions. Applying this model to the SessionGraphs approach has three advantages: First, it provides a structure compliant to all tasks the user works on; second, it can be used to externalize and organize this structure of tasks and the history of related activities; third, it can be used to sub-divide the growing history into small, manageable, and meaningful units, further below introduced as single sessions.

An example will illustrate how the HSO can be applied to structure a set of web-related activities as they could occur in someone’s daily work (see Figure 2-3). The numerical classification is used only to clarify the nesting of the tasks and sub-tasks. It will be shown later in the text how these activities could be executed using both a regular browser and the SessionGraphs tool (pages 92f).

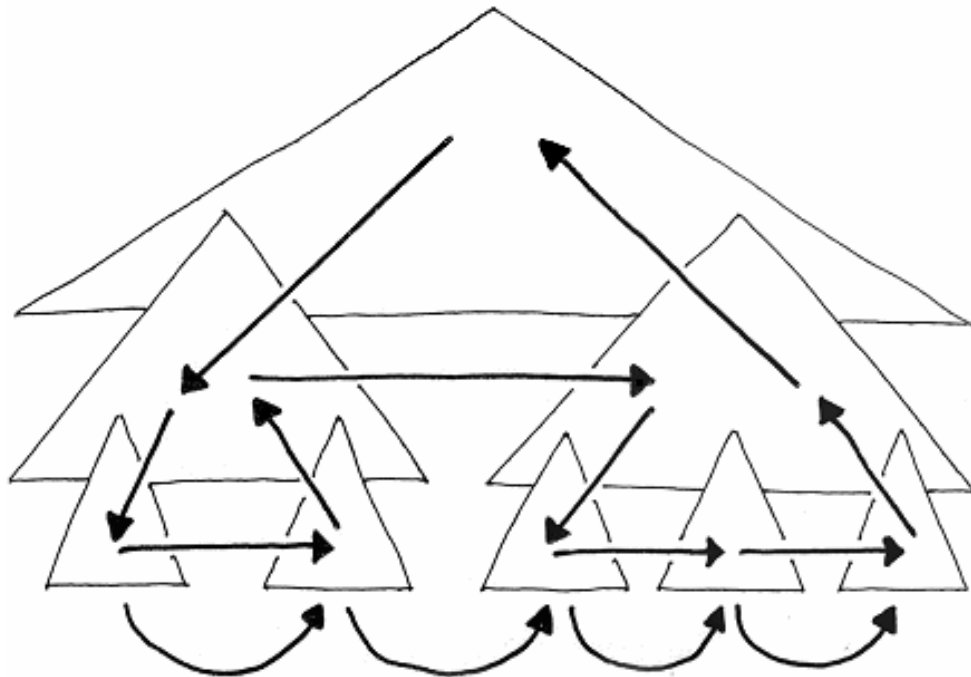


Figure 2-2: Tasks & hierarchical-sequential structure (cf. Volpert 1994: 20). Each task (triangle) is associated with its own sub-task, forming a hierarchy of tasks. Straight arrows symbolize plan creation; round arrows depict the resulting sequential temporal execution of activities, as defined by the deepest sub-tasks.

Example for task structure according to HSO: University professor Hera is preparing her information visualization class for the upcoming summer (compare Figure 2-3). She considers the necessary activities to fulfill this task: clarifying the goal of the class (1.1), viewing new projects and publications (1.2), and preparing the website for students (1.2). Another unrelated task she has to work on is to prepare a trip to Sweden (2). This involves finding a flight and a place to stay.

Two unrelated tasks are executed over time in this example. Sub-tasks overlap in the sequence of execution. Straight lines in the figure represent the logical hierarchy; round arrows represent the sequential task execution. Round arrows always connect leaf nodes, which are operations that Hera can do without further planning. In the SessionGraphs approach they would represent single sessions. The sub-tasks of preparing her class and planning the trip to Sweden would be modeled similarly as tasks.

Several distinctions are presented next to characterize information gathering tasks in more detail.

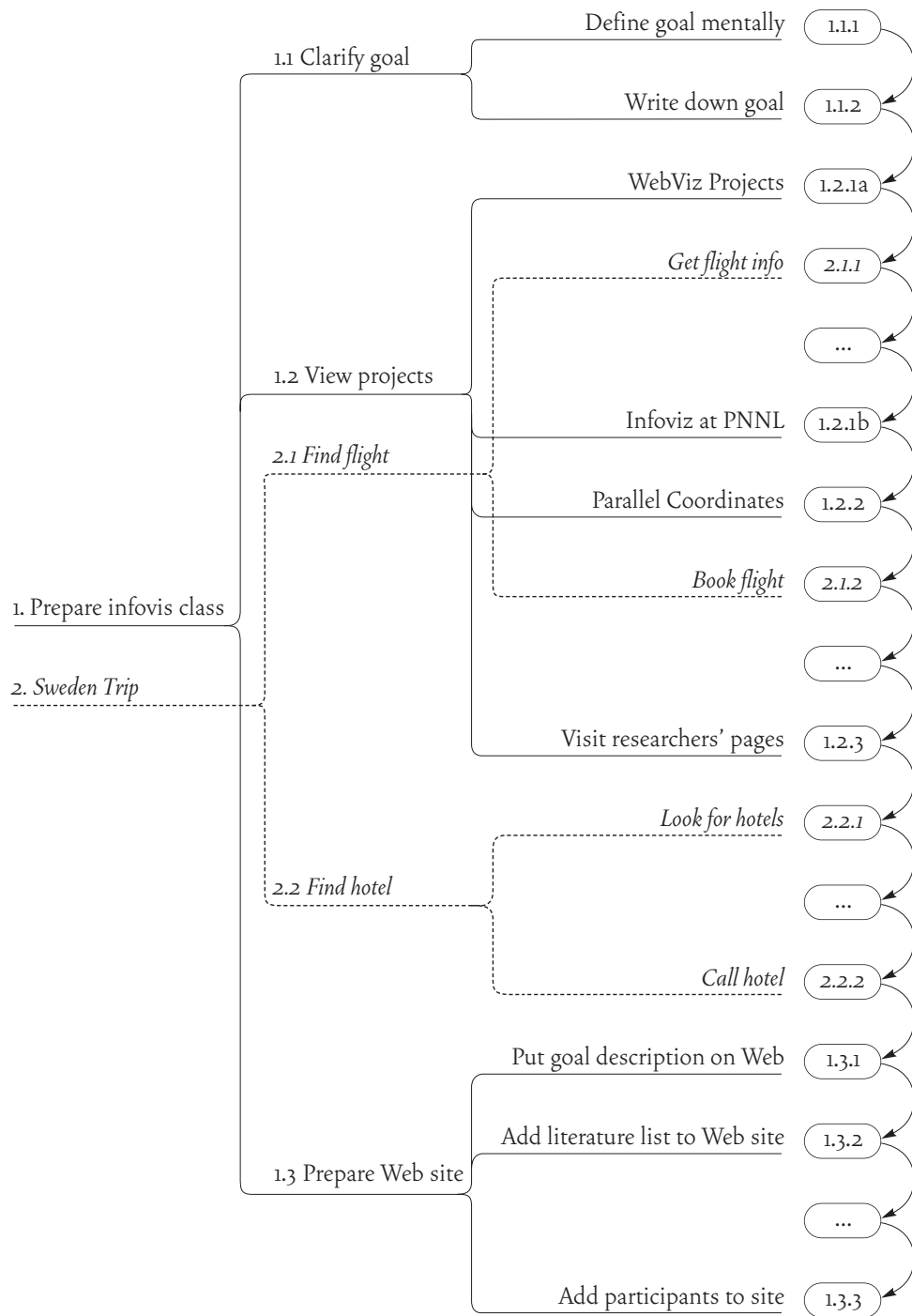


Figure 2-3: Example of a structured task.

2.2.3.2 Information Gathering Tasks

In accordance with ART, information gathering tasks are commonly defined as activities characterized by the desire to close a present knowledge gap (Krikelas 1983: 6; Marchionini 1995; Belkin 1980; Dervin 1983) – as tasks that aim to answer a question or solve a problem (Marchionini 1995: 36). More specifically, *web-based information gathering* refers to any purposeful activity that utilizes resources or services on the web to change a person's state of knowledge (Marchionini 1995: 5).

In general, one can distinguish between strongly- and weakly-structured tasks (Oberquelle 1994: 98f). Whether an information gathering task is strongly- or weakly-structured, basically depends on its *directedness* – which may range from very unspecific to precisely defined. Aguilar et al. (Aguilar 1967; Daft & Weick 1984; Weick & Daft 1983) distinguish common seeking strategies in the non-web context by increasing directedness, which can be extended to the web domain. According to them, *undirected viewing* refers to the exploration of a broad field without a highly specific purpose – the user maximizes sources and minimizes the costs of time, money, and energy (see also Marchionini 1995: 38). This is the least structured strategy. *Conditioned viewing* addresses a more clearly defined area of interest (Aguilar 1967: 20) – the user collects objects for further investigation, visits less items and compares them more accurately. Correspondingly, two strategies of search describe a more active behavior, guided by a clear goal, and a clear idea of how this goal can be achieved (Marchionini 1995). *Informal search* denotes the limited retrieval of specific information without using pre-established formal methods (Aguilar 1967: 20). *Formal search*, finally, follows a pre-established and well-structured plan to retain comprehensive information. It is used for highly important tasks such as developing well-founded decisions and for strategic planning. All these activities also appear on the web and influence navigation behavior.

The more specific the search strategy is, the smaller is the number of sources, and the higher is the accuracy and willingness to pay a higher price in terms of time, effort, and money. Viewing is used to gain an overview, to monitor processes (e.g. to stay abreast of a certain topic), to clarify information needs iteratively and to learn serendipitously, since the environment invites such behavior (Marchionini 1995: 102ff). Viewing demands the continued attention of the seeker and is hard to delegate. Analytical, well-structured searches, so the authors, can more easily be transferred to professional intermediaries who are trained in advanced querying to optimize recall and precision – this, however, should hardly be an option for a web user.

A complementary, valuable perspective to web-related behavior is provided by information foraging theory (Pirolli & Card 1999) that helps to understand web users as they navigate between a variety of websites until finally closing their knowledge gap – or giving up (*ibid.*: 9). It compares websites metaphorically to “bushes”

(*information patches*) that contain valuable “berries” (*information*). After the user has “eaten” several “berries” on one site (“bush”), it may be more profitable to pay the costs of finding the next promising website (“bush”) than to spend more energy and time in seeking the last “berries” on the current site. The higher the directedness of search is and the higher the *primedness* is concerning the location (Darken & Sibert 1996b)¹, the fewer sources and “bushes” will be visited. For revisitations, this theory provides a plausible model why, in many cases, people would only spend limited time on one “bush” trying to revisit a certain “berry” before trying to serve their information need on another “bush”.

A final helpful distinction is based on the non-web-related research by Ellis et al. (1989; 1993; 1997) who observed information gathering behavior in industry. They distinguish between six information gathering activities: *starting*, *chaining*, *browsing*, *differentiating*, *monitoring*, and *extracting*.² The combination of Ellis’ and Aguilar’s approaches is transferred to the domain of the web by Choo et al. (1998; 2000a; 2000b) in order to identify corresponding activities and revisitation patterns during information gathering.

With respect to revisitations, the aforementioned distinctions can help to clarify the situations in which revisits occur (see also Obendorf et al. 2007). In the short-term, revisits are likely during undirected viewing due to *backtracking* processes mainly in hub and spoke navigation patterns – i.e. during browsing and differentiating according to Ellis et al. Similarly, revisits are likely in directed search, e.g. when the user revisits search results lists or reformulates queries. In the medium and long-term, revisitations are more often motivated by the task structure. People need to *resume* earlier sessions, they *re-utilize* web applications or they *observe* a certain page to find out about modified content (monitoring and extracting according to Ellis et al.). In the long- and very long-term, revisits also become likely when people try to *rediscover* certain content of special importance.

Before further detailing the notion of revisitation, a theoretical approach to sessions is presented, since they build the smallest units of meaningful web-related activities in the SessionGraphs approach.

2.2.3.3 Sessions

Using the web for a certain purpose typically results in a sequence of lower level activities, such as following links, using search engines, or filling out forms. Each sequence of such activities that builds up to a meaningful unit in the user’s mind could rightfully be called a *session*. However, many different more formal definitions exist about what the term *session* should exactly mean. Since web usage research is

¹ A user is *primed about a target* when he knows the target’s location in advance.

² A similar distinction is provided by Marchionini (1995: 59).

largely driven by the interests of providers of mainly commercial web sites who want to know what their customers do in order to optimize their site's usage and their commercial profit, it is not astonishing that the notion of a *session* in the context of web usage is mostly constrained to activities related to one specific web site. Accordingly, Berendt et al. define a *real session* as “the sequence of activities performed by one user during one visit at [one specific] site” (Berendt, Mobasher, Nakagawa & Spiliopoulou 2002: 160, parenthesized supplement by author). Employing this provider-oriented perspective in order to optimize the usage of single websites is certainly logical and sensible. Many technologies, mostly cookie-based, were developed to implement so-called *sessionizing heuristics* (Berendt et al. 2002) that support these efforts by automatically grouping those visits that are likely to belong to a single user and a single phase of visits. Commonly, cookies are used to identify a user, and an inactive period of 30 minutes is used as a heuristic to distinguish between different sessions of the same user. This strategy produces reasonable results for site providers in order to analyze their visitors' behavior and to fine-tune their offer (this topic is further elaborated by Mobasher 2006; Facca & Lanzi 2005).

However, applying this provider-oriented perspective on sessions to the design of client-side history tools that accompany a user throughout *all* of his or her web-based activities across a multitude of different websites, creates rather constrained solutions that do not serve the user's needs adequately. Defining sessions as within-website activities is an unnecessary limitation of a history's capabilities.

Consequently, to serve a more user-oriented perspective, sessions in history tools must be able to include all between-website navigations that cohere in the user's mind into a single meaningful group of activities.

A thoughtful description of a user's activities during the usage of a hypertext is provided by Jim Rosenberg (1996). He established the terms *acteme*, *episode*, and *session* to describe the process of reading closed narrative and literary hypertexts. With some modifications these terms can be transferred to the web: *Actemes*, then, denominate lowest-level activities, including link-following, querying, backtracking, scrolling, and the single interactions created by using a web application. *Episodes* label the next higher-level entity: “An episode is simply whatever group of actemes cohere in the reader's mind as a tangible entity” (Rosenberg 1996: 23). Figure 2-4 shows an illustration of an episode. Different lexia are connected by link-following actemes and result in a single coherent episode.

Rosenberg reserved the term *session* to denominate longer phases of contiguous activity in closed, mainly narrative hypertexts, comprising several episodes or attempts of striving for episodes. Ending a session – in Rosenberg's sense – means to stop reading the hypertext (Rosenberg 1996: 22). On the web, however, this is not necessarily the case. The user may switch to another, completely unrelated activity, but still use the web. Thus, the term *session* will be used in this thesis to comprise all those episodes that a user executes on the web which are related to *one common*

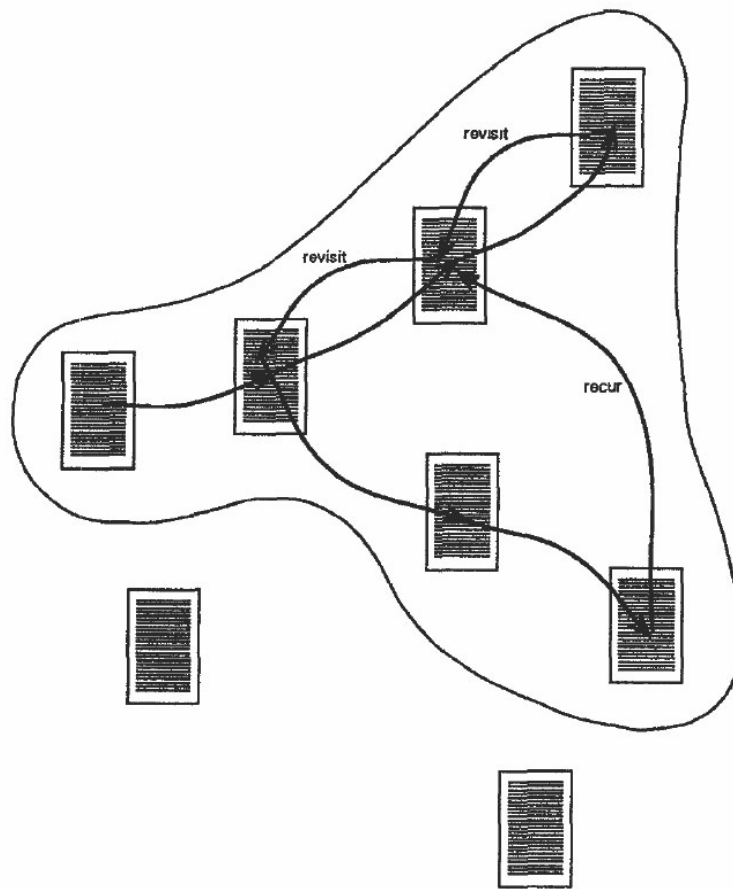


Figure 2-4: Towards a notion of session - Rosenberg's episode (1996: 25).

Rosenberg depicted lexia as atomic entities connected by actemes of link-following. The curve determines the episode's border.

goal – be it precisely defined or rather vague. Only the user can tell which visits belong to a certain session, since only he or she knows about the underlying goal. This notion of a session will also be referred to as an *intentional session*¹. An intentional session is distinguished from a *technical session* (in short *T-session*), which refers to the heuristically identified sequence of visits as an approximation of an intentional session. The heuristical identification of technical sessions will be used to subdivide user log files into potentially meaningful pieces, as well as to offer semi-automatrical session support in the SessionGraphs prototype. The term *SG-session* is used to refer to the constructs provided by SessionGraphs that enable the user to model and maintain intentional sessions externally. The terms *SG-session visualization* (or session visualization) and *SG-task visualization* (or task visualization) will be used to refer to the visual externalizations.

¹ This term is congruent with Berendt's term *real session*, without the restriction to a single site. If not specified more precisely, the term *session* will denominate an *intentional session* in the following.

Rosenberg considers different reasons why users end a session. They are likely to happen on the web as well and vividly show why it is important to provide good support for session resumption. After *ending by accident*, an external cause such as a phone call, power failure, or browser crash, it is likely that the user will try to continue later. Also when *ending by satiation* the user is only currently sated with related activities, but will probably start again later. A third comparable situation is given when *ending by fatigue* – when no interesting episode is perceived at the moment. In all these situations the user probably wants to resume a session later. Only when *ending by satisfaction*¹, when a “success point” is reached by accomplishing the session’s goal, it is not necessary to resume the session later. But still then it may be valuable to store session information, for example to reuse it in similar sessions in future. In at least three out of the four situations it is advisable to provide proper means to resume the earlier session and possibly revisiting several of the earlier visited pages again. Even if O’Conaill and Frohlich (1995) found that 41% of the time people do not resume their original task after an interruption, they should be supported as good as possible when they want to resume it. Such resumption support is not adequately given today, as shown below.

Rosenberg concludes by demanding a “formal gathering interface” that assists the user in executing episodes and sessions; a tool that also supports several concurrent activities and that visualizes the user’s sessions as they evolve. The SessionGraphs approach provides such an interface for the web. It builds upon the initial vision of hypertext – Vannevar Bush’s concept of trails as formulated in his Memex concept (see also page 222).

2.2.4 Defining Web Page Revisitation

Current literature blurs fundamentally different user motivations and activities of revisitation, ending up using an ambiguous, unclear terminology. One risk thereof is to overlook numerically minor kinds of revisits that are still meaningful to the user and important to support. The following nomenclature and taxonomy serves a clearer discussion in the future.

Address Based Definition

Following the spatial metaphor underlying many aspects of the web (see also chapter 2.3.2.1), the term *revisit* (or *revisitation*) will be used here to describe the repeated

¹ Similar to Douglas’ concept of closure (1994), in which a session is likely to be finished. The hypertextual experience exceedingly depends on the author’s aptness to create an environment in which the user is likely to have satisfying episode experiences.

visit to a *web page* as identified by its *location*, i.e. its *address (URI)*.¹ Thus, a revisit may refer to the same or modified content. This use is consistent with earlier research. Looking at the topic of revisitations from a user's perspective, to some right also actions such as bringing to front a hidden window, selecting a hidden tab, or even scrolling to a certain position in a long document, could be considered as revisits – since the user wants to review content that was presented earlier. To keep the nomenclature compliant with existing research, however, these cases will not be considered as revisit here.

History Systems

Regarding the tools that support the user in recurrent processes, one can distinguish *transition-based* and *state-based* history systems (Lee 1992: 29). The difference is that a state-based system records and recovers document information rather than navigation or action information. *Bookmarks* are obviously state-based, but also *back* and *forward* are state-based, since they push and pop URLs on and off the stack, not navigation actions. In contrast, *web-based applications* can also require *transition-based history systems* that record the interface actions that lead to changes of the system's state rather than the states themselves. Transition-based systems are appropriate where similar *actions* should be applied to different *objects* (e.g. in image editing tools, word processors, etc.). History systems that support revisits on the web should be *state-based*, since certain documents should be recovered, independent from the navigation actions that lead to their visit. Many web-based applications would need an *undo* function offered by a transition-based system, separated from the back function (Weinreich, Obendorf, Herder & Mayer 2006b; Obendorf et al. 2007). If *undo* and *back* are not separated it may lead to confusion. This thesis focuses on state-based history systems that support the user in navigation between single web pages. It does not consider history systems for web applications or for special forms of hypermedia – such as video, audio, or online games that require extended or different history treatment.²

Following this state-based interpretation, in this thesis the term *history* denominates the sequence of all web pages a certain user has visited on the web (compare Lee 1992: 11). Consequently, a *web history tool* or *system* is a piece of software that simply uses these technically recorded histories to support the user in his or her upcoming activities by making revisitations easier. Each history designer and/or researcher has then to define which items to include and which to exclude

¹ In this definition, a *visit* refers to the process of loading a web page into a browser window and the term *web page* denotes any resource on the web, no matter how it is encoded (HTML, PHP, XML, PDF, MP3, Java, etc.).

² More information on non-web history systems such as command line interfaces and direct manipulation systems can be found in (Linxi & Habermann 1986; Lee 1992; Greenberg & Witten 1988; Greenberg 1993; Tauscher 1996a).

Dimension	Types of revisits in this dimension	
Temporal occurrence	Short-term	< 1 hour
	Medium-term	1 to 24 hours
	Long-term	> 24 hours
	Very long-term	> 1 week (special type of long-term)
Purpose	Content revisit	Review same content revisit View modified content revisit (monitor / observe)
	Navigation revisit	(to intermediary navigation page)
	Tool revisit	(reuse)
	Non-intentional revisit	
Session / Task context	Intrasession revisit	
	Intratask but intersession revisit	
	Intertask revisit	
	Special cases:	
	– Parallel session revisit (an intersession revisit) – Session resumption revisit (an intrasession revisit)	
Arity of target(s)	Single-page revisit	
	Multi-page revisit	

Table 2-2: Taxonomy of revisit types along four dimensions.

from the history; it has to be considered which items are defined as artifacts (ads, frame pages, etc.) and how to keep the history clean from such artifacts. It also has to be decided which additional information should be stored – such as window IDs, timestamps, and content related information.

Taxonomy of Revisits

The taxonomy in Table 2-2 presents a taxonomy of various revisitation types along four different dimensions. Revisits are distinguished on the temporal dimension according to the interval that has passed between two subsequent visits to the same URI. On an objectively reasonable basis, no temporal boundaries exist. Thus, distinctions will be used that are culturally meaningful: *short-term revisits* (within 1 hour) are distinguished from *medium-term revisits* (between 1 and 24 hours) and *long-term revisits* (after 24 hours). Additionally, those long-term revisits that occur after one week are labeled *very long-term revisits*.

Along the dimension of purpose – or the user’s intention – *content-*, *tool-*, and *navigation* revisits are distinguished. A fourth type, *non-intentional* revisits, denotes revisits that actually took place but were not intended – so those that happened by mistake or accident. Only the user can tell to which of these types a certain revisit

belongs. *Content revisits* are defined by the intention of reading, modifying, or creating content on the web. By a “*review same content revisit*”, the user wants to review exactly the same content as before (e.g. product information, contact information, any kind of detail that was forgotten in the meantime or not read the first time, etc.), whereas a “*view modified content revisit*” is characterized by the expectation of finding modified content compared to the last visit (e.g. news sites, weather information, conference portals, etc.). A *tool revisit* refers to a situation in which a user reuses a web-based tool or service (e.g. search engine, train schedule, city map, or internet shop). Such tools mostly contain forms where the user can submit specific information. Finally, a “*navigation revisit*” refers to the visit of an *intermediary navigation page C* that is primarily motivated by the desire to reach a certain page B from a page A, but not to view or use the content of C. Navigation revisits occur, for example, during link following where the user scans A for promising links, follows the most promising ones over and over again, thereby (re-)visiting several pages C_1, C_2, \dots until finally reaching B or giving up. Scanning intermediary navigation pages consumes valuable resources such as time and attention – it increases cognitive overhead; entire pages have to be examined to find the promising link for the next step of navigation; furthermore, the user has either to actively ignore potentially interesting detours or to weigh the costs of following them compared to directly heading towards the target (compare Conklin 1987). Also during backtracking additional cognitive effort is necessary, when intermediary navigation pages are not scanned for links, but instead for evaluating whether the target page has been reached. Thus, history tools should try to minimize navigation revisits by providing more direct ways for reaching a desired target page.

According to a revisit’s session and task context, one can distinguish three different types. First, *intrasession revisits* are visits to pages that were already viewed in the same session before – these are necessarily also intratask revisits. Second, *inter-session, but intratask revisits* to pages that were already visited in the context of the currently active task, but not during this session. And third, *intertask revisits*, which have been visited before, but not in the context of the same task. These distinctions will be used to define the cost hierarchy for history accesses in the SessionGraphs approach (see page 82). Two special cases are *parallel session revisits* and *session resumption revisits*.

Finally, the *arity* dimension addresses the number of target pages that a user intends to revisit (compare with the discussion of quantity by Marchionini 1995). For a *single-page revisit*, the user intends to only revisit one single page at a time. A *multi-page revisit* is defined when the user initially already intends to revisit several pages – be it sequentially in the same window or in parallel distributed over several windows. Multi-page revisits are likely to occur during continued sessions, comparisons, and repeated tasks that require several sources or tools to be solved.

The taxonomy presented above identifies important and helpful distinctions that will be used in the SessionGraphs approach. Further dimensions are also possible (place of execution – i.e. same browser, same computer, different computer, etc.), but will not be considered here.

2.2.5 Status Quo of Revisitation Behavior Analysis

In addition to the mentioned theoretical approaches, revisitation behavior of users of information systems was also analyzed as it actually occurred in practice. Although many studies have investigated various aspects of web-related behavior, only a few of them addressed the topic of revisitation. One reason might be the difficulty in finding participants who are willing to share such intimate data over a considerable period of time – like weeks or even months; we experienced this in our own studies (Weinreich et al. 2006b), compare also with (Pitkow 2000). Furthermore, in most setups they have to be willing to install some kind of study software on their computer, since the data needs to be logged on client-side. Finally, frequently used server-side statistics cannot be utilized since they only cover activities on one domain and do not include details about browser interaction.

The existing knowledge about web-based revisitation was mainly obtained in three extensive long-term client-side studies conducted by Linda Tauscher and Saul Greenberg at the University of Calgary, Canada, (Tauscher 1996a, 1996b; Tauscher & Greenberg 1996, 1997a, 1997b), Andy Cockburn and Bruce McKenzie at the University of Canterbury, New Zealand, (McKenzie & Cockburn 2001; Cockburn & McKenzie 2001), and Lara Catledge and James Pitkow at the Georgia Institute of Technology, USA, (Catledge & Pitkow 1995).

The data gathered in these studies are now eight (1999) to thirteen (1994) years old. Since then, the web's technology, offers and users' behavior have changed. The web has been developed to a much higher degree than when the results of these studies were publicized.

Catledge and Pitkow (1995) analyzed the log-files of 107 users - staff, faculty, and students at the Georgia Institute of Technology's Computing Department. The logs contained information comprising about three weeks of interaction with the XMosaic 2.4 web browser gathered during the summer of 1994, resulting in a total of about 31,000 navigation actemes. They found a mean visit rate of circa 14 pages per day. Analyzing the interaction with the browser they revealed a high usage of hyperlinks (about 52% of all interactions), followed by a still frequent use of the back button (41%). In contrast, the use of the forward button and the bookmark system (at this time called »hotlist» with limited capabilities, see Shen, Kwok & Yang 1996) only accounted for about 2% each. The high usage of links and the back button indicated many short-term revisitations. This was confirmed – they found

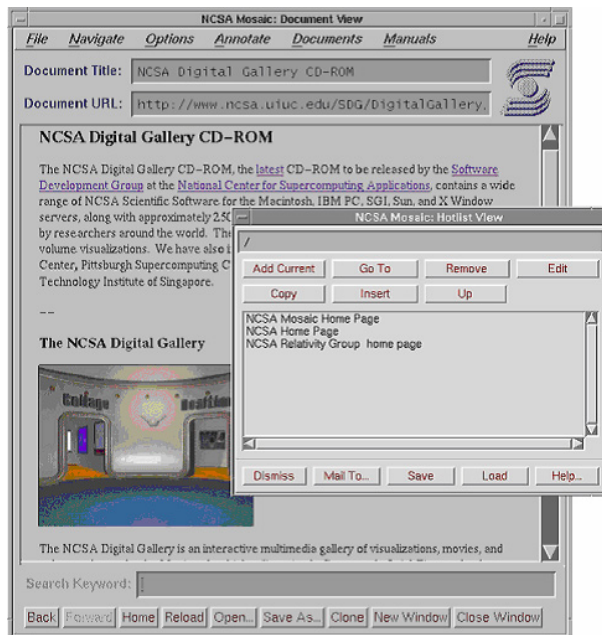


Figure 2-5: NCSA XMozilla 2.6 browser with Hotlist View.

This browser was used in the Tauscher study (1997a). Back and forward buttons (at the bottom of the window) did not yet provide jumps over several pages. The hotlist (a kind of bookmark list) supported neither thumbnails nor drag-and-drop interaction (NCSA 1997). Hierarchical bookmarks were possible only after version 2.5.

very frequent hub and spoke navigations. Users kept personal home pages with link lists as indexes to interesting locations on the web.

One year later, during five to six weeks in the winter of 1995, Tauscher and Greenberg studied 23 subjects, also computer science staff, faculty, and students, as well as telecommunication engineers and programmers (Tauscher 1996a; Tauscher & Greenberg 1997a). Using the XMozilla 2.6 browser (see Figure 2-5), they collected around 19,000 navigation actions and found a mean visit rate of 20 pages per day and user. Again, the high amount of links and back button actions was confirmed (about 42% and 30% respectively in this study). They introduced the *recurrence rate* of URLs in the log-files, i.e. the ratio of revisited pages among all visits, as a general measure for revisitation (Tauscher & Greenberg 1997a: 112). It can be interpreted as the probability that the next page visited was already visited before.

$$\text{Recurrence Rate } R = \frac{\text{total number of URLs visited} - \text{number of different URLs visited}}{\text{total number of URLs visited}} * 100\%$$

The mean recurrence rate among their subjects was 58% ($\sigma = 9\%$). By re-analyzing Catledge and Pitkow's data they found their recurrence rate to be 61% ($\sigma = 9\%$). This meant that three out of five pages had been already seen before, an unexpectedly high amount.

Cockburn and McKenzie (2001) re-examined the situation in the winter of 1999/2000. The log-files of 17 users, all computer science oriented, were analyzed, covering four months of Netscape Navigator usage (versions 4.5 to 4.7), approximately 85,000 navigation actions in total. In this case, a slightly different methodological approach had been chosen. Their participants used their unmodified, regular browser and were unaware of the study while the data was being collected. The subjects were asked at the end of the study for permission to use their daily backups of the Netscape `history.dat` files. This approach, therefore, excluded the problem of “Hawthorne Effect” modifications (Mayo 1933: 59), since their subjects were not aware of being observed. The drawback was that this also meant that no detailed information would be available about specific user interface interactions, since the data were of lower resolution - only the timestamps of every day’s first and last visit to a URL was logged, not the dates inbetween. The recurrence rate was even higher, 81% ($\sigma = 10$) and the mean daily page visit count doubled to 42 visits.

Beyond the unexpected high recurrence rates, these studies revealed that most revisits occur in the short-term following a power law distribution with extreme recency and rapidly declining values for growing temporal distances. In Tauscher’s sample, the chance that any visited URL was contained in the previous six URLs was almost 40%.

A complementary perspective was chosen by William Jones, Harry Bruce, and Susan Dumais, who investigated how people Keep Found Things Found (KFTF – Jones, Bruce & Dumais 2001, 2003; Jones et al. 2002; Jones 2004; Bruce, Jones & Dumais 2004). Rather than analyzing log files, they observed and interviewed participants in several studies in order to examine long-term revisits more thoroughly. They discovered that several non-browser-based “workaround” strategies had been used for revisitation rather than the intended history mechanisms that were integrated into the browser – a vivid demonstration that appropriate support is still lacking (Jones et al. 2001). For example, several participants of the KFTF studies sent an email containing a URL to themselves, along with notes on relevance and actions to be taken. Some also emailed addresses to someone else – subsequently asking the recipient or searching the sent box in order to re-obtain the address for themselves. Further “workaround” techniques included printout on paper – some even wrote URLs on sticky-quotes and attached them to their monitor. Some saved URLs as a file, sometimes by pasting several addresses into one document or adding them to a personal web site. A common technique was to do nothing – and to count on finding the page later again via a search service or via direct URL entry. Still further strategies included writing the address on paper, and, the use of bookmarks. They reported that each “keeping method provided a range of functions but none of the observed methods allowed for all desired functions” (Bruce et al. 2004: 1), indicating a special shortcoming of the current situation. In their “re-finding” study with thirteen participants, they report that more than 95% of all attempts to find

something again after three to six months were successful, and that about two thirds of the applied re-finding methods did not require any explicit keeping behavior.

These findings could lead to the incorrect assumption that for long-term revisits, history tools are, first, not as necessary as potentially assumed, and, second, that the applied techniques fulfill their needs quite sufficiently. This, however, would be a premature conclusion: it should be noted that the study only involved revisits to websites in which the subjects themselves had already initially estimated the likelihood for a long-term revisit within the next twelve months at over 75%. Thus, from the outset this study only considered the less-problematic cases, i.e. those containing relatively important web sites where the subjects know in advance that they will probably come back later. Moreover, it is very likely that during the first visit the subjects then made sure that they would be able to come back again, e.g. through bookmarking (a page) or by ensuring that they had memorized a particular URL. Therefore, this study fails to address one of the more severe shortcomings surrounding bookmarks, which is well expressed by Benjamin Bederson who is quoted in a New York Times article (Guernsey 2004) stating “that bookmarks ‘are pretty hopeless.’ The concept is flawed, he said, ‘because it assumes in advance that this is a page that you want to revisit, and you don’t always know that’”. Exactly these problematic situations, however, are excluded a priori by the study’s design, as are all situations in which the probability to return is lower than 75%. In addition, the authors report that the time required for a revisit varied largely – that the median, however, was already circa one minute, just to revisit one specific web page – and that some trials took as long as five minutes (Bruce et al. 2004). In contrast to the initially suggested assumption, this even underlines that current tools fail in the ability to appropriately support highly likely long-term revisits. Finally, the authors report that, in spite of the high overall success rate, “participants occasionally expressed frustration at not knowing where to ‘go’ in order to re-access a web site” (ibid.) and that 6% “of the recall trials involved participants using three or more methods in order to access a particular web site”. They vividly underline these findings with an example: “one information specialist used the following sequence of methods before finally re-accessing the site: 1.) try to re-find from another web site, 2.) use search service, 3.) try direct entry of URL, 4.) look in paper file folder; before successfully re-finding the website [...]” (ibid.). Even this trial and error method was counted as a success, although it clearly proves the inadequacy of several revisitation strategies and tools.

In conclusion, the KFTF studies provide highly interesting insights on re-finding strategies currently in use, revealing the large hodgepodge of “workarounds” and methods that are partly diverted for long-term revisitation purposes. When carefully interpreted, these studies underline the urgent need for improved long-term revisitation support.

The aforementioned insights lead to two conclusions regarding the present work: If revisitation is such an important part of web navigation, it needs to be supported as well as possible. Current and suggested approaches of history support and their problems will be presented next. On the basis of the evaluation of their strengths and weaknesses, the SessionGraphs approach was designed. Second, the three clickstream studies are too old to mirror current behavior on the web, which underwent a variety of transformations both technically and in terms of usage since then. New insights were overdue. Thus, we executed a fourth study to update and refine these findings. Its relevant aspects are presented in chapter 5.

2.3 Status Quo of History Tools

This chapter presents currently available history mechanisms, as embedded in today's common web browsers and as suggested in history research projects. It summarizes their functionality and addresses their benefits and shortcomings.

2.3.1 History Support in Current Web Browsers

Microsoft Internet Explorer is still by far the most commonly used web browser (compare Table 2-3 and Figure 2-6), mainly in its version 6 (TheCounter 2007). The reported share of Internet Explorer in February 2007 ranks between 71% and 85%.¹ Since 2004, Mozilla Firefox has become very successful and is expected to gain more of the Internet Explorer share in the near future (current global share 12%, almost 30% in Germany – Adtech 2007).² These two major browsers are followed by Apple's Safari, Opera, and Netscape.

Web Browser	TheCounter 02 / 2007	OneStat 01 / 2007	Adtech (Europe) 02 / 2007
Microsoft IE	71 %	85 %	78 %
Mozilla Firefox	12 %	12 %	14 %
Netscape	12 %	0 %	5 %
Apple Safari	3 %	2 %	2 %
Opera	1 %	1 %	1 %
Others	1 %	—	—

Table 2-3: The most popular web browsers in July 2006.
They provide very similar history and navigation support (source: OneStat 2006).
Sources: (TheCounter 2007; Onestat 2007; Adtech 2007).

¹ No method currently exists for determining exactly how many people use what kind of browser. Usually numbers are published by companies that distribute website analysis software. They can detect the kind of browser software their customers' sites are accessed with. Onestat (www.onestat.com) e.g. has about 50,000 websites in 100 countries as clients. Accordingly, the published numbers are always regionally biased and influenced by the specific users that visit the monitored sites.

² Niels Brinkman, co-founder of OneStat.com, at:
http://www.onestat.com/html/aboutus__pressbox36.html

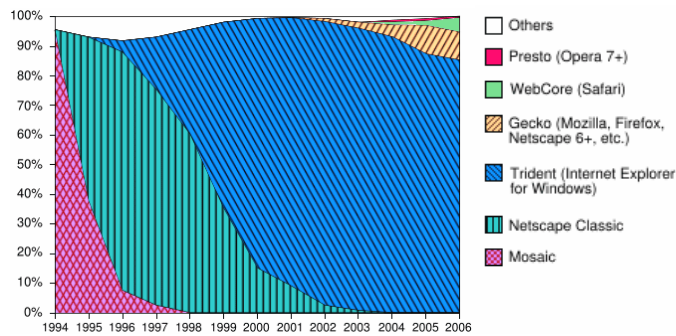


Figure 2-6: Usage share of web browsers, 1994-2006 (Wikimedia 2007).

Current web browsers (see Figure 2-7) usually support the following history techniques: Back and forward buttons – with attached pulldown menus (1), URL entry field with autocompletion – and an attached pulldown menu on the right (2), Google toolbar extension (3), Toolbar bookmarks including folders (4), Home button (5), bookmarks and history lists (both accessible via the menu or the sidebar), breadcrumbs (color-coded anchors that lead to visited pages), as well as save- and print-functions (via menu).

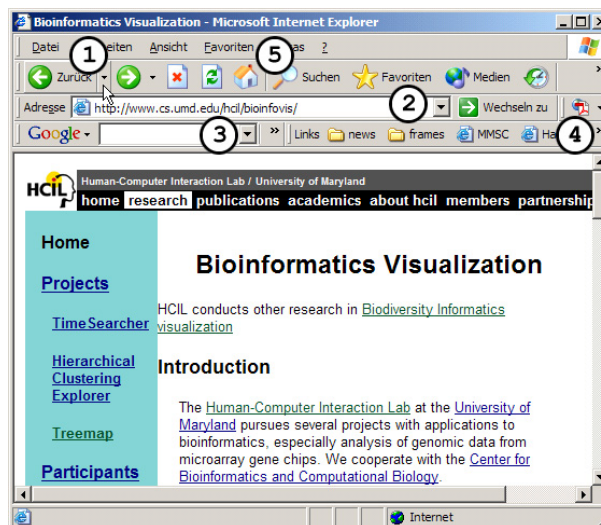


Figure 2-7: Microsoft Internet Explorer 6.0 with history & navigation tools.

2.3.1.1 Backtracking

The navigation back along an earlier path, backtracking, is the second most frequent interaction technique after hyperlink usage (Tauscher 1996a; Weinreich et al. 2006b). It is often motivated by the desire to explore alternative branches in the

hypertext (Nielsen 1995b: 245). Most browsers offer back and forward buttons, recency ordered pulldown menus¹, a “navigation” menu, and keyboard shortcuts, e.g. “ALT” + “←” on Microsoft Internet Explorer for going back. Some browsers support gestures, e.g. Opera since late 2000 in version 5 or, for Mozilla browsers and Internet Explorer, plugins are available such as “Easy Go back”. According to Fitts’ law (Fitts & Peterson 1964, 1954), gestures save valuable time in comparison to buttons. Earlier interface suggestions for backtracking include pie menus (Hopkins 1991; Callahan, Hopkins, Weiser & Shneiderman 1988) and marking menus (Kurtenbach & Buxton 1993). Hardware backtracking support includes special mice and keyboard systems. Considering the immense share of back events in relation to all interface events, any improvement is valuable.

Backtracking algorithms were originally developed for the solution of combinatorial problems (Bitner & Reingold 1975)², inspired by the Greek myth of Ariadne who helped Theseus to escape Minotaur’s maze. They try to extend partial solutions to an overall solution while keeping a record of the path. When a dead-end is encountered they backtrack to an earlier partial solution.

Without backtracking, any hypertext interface would be highly unusable – it is an essential ingredient to the non-sequential structure of hypertext. To visit a target B at a distance greater than one from a current node A, the user has to visit some intermediary nodes, similar to Theseus in the maze. Ideally, the shortest path is chosen. The user, however, normally just sees one level of outgoing links, like Theseus who only saw the current room’s adjacent doors. Thus, a hypertext user is likely to reach dead-ends that are useless for the current task and to track back to earlier nodes that promise better alternatives.

Backtracking can be described as a tree search. When a fork in the tree is reached, one of the branches is searched downwards, following depth first search. If the desired target is not found along this branch, another branch will be explored and the path information concerning the last inspected branch will be dismissed (Bitner & Reingold 1975). Backtracking algorithms only use one list of open nodes, implemented as a LIFO stack, storing the (partial) current path.

Current browsers use exactly this stack-based technique. In contrast to solving combinatorial problems, however, the loss of path information can cause problems here. The navigation sequence A-B-C-D <back to> C <back to> B <follow link to> E causes C and D to be popped off the stack. They can no longer be accessed using the back function, even if this may be a desirable behavior after misleading navigations or detours. Often this happens unexpectedly and leads to problems, e.g. during hub and spoke navigation. People will have difficulties at some point in predict-

¹ Pull-down menus for backtracking were introduced by Netscape Navigator 4 in 06/1997.

² Problems solvable by backtracking can be formulated like “is there a way to ...”, “how many ways are there to ...”, “list all possible solutions to ...”.

ing what page the back and forward buttons will cause to appear (Greenberg, Ho & Kaasten 2000). Rather than having an internal model of how the back button works, they apply a “click-until-reached” strategy (*ibid.*).

Several alternative algorithms have been proposed (Greenberg & Cockburn 1999; Nielsen 1995b): Pure recency lists withdraw since they oscillate between two pages after repetitive back usage. In order to receive well-working recency lists backtracking events have to be treated differently than the remaining navigation events. Greenberg and Cockburn (1999) suggested a temporally ordered list with duplicates stored in their latest position. Tauscher and Greenberg (1997a: 124ff) evaluated hierarchically structured approaches that showed much better performance values, according to a theoretical analysis. However, the authors themselves mention serious doubts in the value of these theoretical results, since the high cognitive efforts necessary to navigate the hierarchical menus were not regarded in their evaluation.

It is still unclear which algorithm is best for a browser’s back function. One possibility is to enable the user to choose between losing and not losing information when going back. Bieber et al. (1994: 165) rightfully ask: “Should backtracking trigger an ‘undo’ operation or simply reflect the current state of the departure nodes?”. Rosenberg (1996: 22) replies: “The simple act of going back may have multiple types. One may revisit a lexia simply to read it again, or it may be a genuine ‘undo’: perhaps the reader didn’t mean to follow that link at all. These are arguably different actemes, though typically not distinguished by hypertext user interface behavior”.

Additional limitations can be found in the restricted number of pages that a user is willing to skip until reaching a target hidden somewhere deeply in the back of the stack; further, the fact that it is not possible to track back across the borders of multiple windows or tabs, and that all backtracking information is lost once a window is closed – i.e. no long-term path history is available. The issue of short-term revisitations that are more and more substituted by the navigation between several synchronously opened tabs or windows is discussed in chapter 5.3.2.

To summarize, backtracking is an essential and frequently used history mechanism that mainly supports short-term revisits, and often does so very well. Its strengths are in its effectiveness, the little screen space it requires, and the little cognitive load due to the “click until recognized”-strategy. Weaknesses include the inadequate support of different intentions for going back, the missing distinction between back and undo, the restriction to short-term revisits in one single window, the incomplete history caused by the stack implementation, and the potential distraction by all the pages presented at full size along the way to a desired target. Alternative approaches should be explored.

2.3.1.2 Bookmarks

Bookmarks (or favorites) are actively set URL-based references for single web pages of special interest. They enable the user to revisit single pages in the long-term. Current browsers offer an editable hierarchy and a toolbar to store bookmarks. Both structures allow storing bookmarks hierarchically in folders. The hierarchy can then be accessed via the menu and the browser sidebar. Referenced pages are usually represented by title and a small picture, the so-called “fav-icon”, which usually only distinguishes different servers. The possibility to differentiate single pages is rarely used. Some browsers (e.g. T-Online) provide thumbnails of the web pages on mouse interaction.

Selecting a bookmark loads the web page referenced by the recorded URL. Bookmarks free the user from having to remember and interpret URLs and from typing them explicitly (Abrams, Baecker & Chignell 1998). As all URL-based techniques, bookmarks bear the risk of invalid links (Cockburn, Greenberg, Jones, McKenzie & Moyle 2003).

More recent developments comprise bookmark archives that can be shared in a work group (like Macropool__Web-Recherche 2007; Inforce__Bookmarks 2007) and tools such as the Firefox Scrapbook extension (Gomita 2007), which allows the local storage of web pages. *Social bookmarking* denotes the collaborative maintenance, tagging, and ranking of bookmarks and is offered by tools such as del.icio.us, Furl, Spurl, Yahoo! MyWeb 2.0, and Magnolia, but also by scientific citation tools like CiteULike and Connotea (compare Gordon-Murnane 2006). *Web syndication*, finally, refers to the integration of frequently updated information into many web pages – technically based on RSS or Atom. RSS feeds, for example, can be integrated into the browser side bar as dynamic bookmarks, e.g. using Firefox's *live bookmarks* feature. They can update themselves and always reference the latest content.

Studies and surveys about traditional bookmarks, as provided until today, reveal different insights: In 1998, still 92% of the 16,000 respondents to the 10th Gvu survey used bookmarks (Gvu 1998a) and most users considered them “essential” for using the web (Abrams 1997: 3-4). Cockburn and McKenzie (2001: 913) found means of 184 bookmarks ($\sigma = 166.2$, 17 collections) and 18.1 bookmark folders ($\sigma = 16.5$) per participant.² Amongst Abrams' users, bookmark archives grew

¹ Image used to identify the organization that a page is authored by (file “favicon.ico”).

² Both numbers excel earlier surveys including (Gvu 1998b) and (Abrams et al. 1998). About 35% of the Gvu participants said, they have more than 100 bookmarks, about 23% had between 51 and 100, and about 30% had 11 to 50 bookmarks. Almost half of Abrams' 23 participants had between 26 and 100 bookmarks, about a quarter had 11 to 25, 14% had 101 to 300, and 2% had 301 or more bookmarks. 10% had only 1 to 10 bookmarks.

with a rate of about one bookmark every five days (Abrams et al. 1998). Our own web usage study (chapter 5) as well as Aula et al. (2005) and the Keep Found Things Found studies (KFTF, page 34) showed that people still use bookmarks frequently.

Growing bookmark archives require additional organization by the users in order to remain in a manageable state. This leads to different reactions and varying degrees of acceptance amongst the users, who weigh the costs of additional work against the benefits of having later revisitation support (Abrams et al. 1998: 44; Pirolli & Card 1995, 1999). This additional overhead is considered to be a severe shortcoming by some of the users. A New York Times interview (Guernsey 2004) quotes William Jones, one of the KFTF researchers, saying “that bookmark lists have become ‘information closets’ that hold a jumble of sites people never return to. Only hyper-organized users sort sites into folders, clean out dead links or click on inscrutable addresses to figure out why they were bookmarked in the first place”. He is further cited: “We say ‘Oh my god!’ and we close the door [...] We don’t like to think that we are that disorganized” (ibid.). Additionally, two respondents of Aula et al. (2005: 587) share this view: “*IE makes it so hard to organize favorites that I leave them all in an ugly pile and don’t rely on them as much as I’d like. Re-org is a pain. The simple tree of the bookmark manager hides nooks and crannies*”.

While this is one perspective – revealing a severe shortcoming – other users weigh the value of an organized archive much higher than the costs of the additional maintenance. Aula et al. (2005: 588) report on ten separately surveyed so-called heavy bookmark users, with archives of more than 500 entries, that they “carefully organize bookmarks, typically with two or three levels of folders. When asked about the successfulness of their bookmark organization, all of them were happy with it. For these people, Bookmarks had become an indispensable tool: *I have spent lots of time thinking about the organization in order to find the ones I need as quickly as possible. I have several folders (and subfolders) named based on the bookmark content, for example Music, Work, Usability, eLearning, Studies, News etc. I’m quite satisfied with the organization - there could be somewhat less folders, though. Yes it is highly successful for my needs over the last ten years! I literally have hundreds of folders. (...) it is a really helpful thing and I would be totally lost without the favorites folder!*”. Furthermore, Abrams (1997) reports that users with large bookmark archives (more than 300) were very willing to maintain them actively. They primarily organized them at the beginning of a session, with only a few at the end. In the GVU survey, between 62% and 75% of all respondents mentioned they were willing to spend the additional effort (GVU 1998a). Concluding, this means a potentially valuable history technique, which requires considerable additional maintenance, is likely to be rejected by some of the users but be accepted and valued by others.

One of the most revealing results of Abrams et al. (1998) is that bookmarks are frequently used as a substitute for session revisitation and resumption support –

instead of a more adequate method, which is still missing: “Users [...] require a means of accessing inter-session history. In the absence of such functionality, users are bookmarking pages to enable access to previous browsing sessions” (1998: 43). Abrams et al. (1998) identified the following uses of bookmarks: to reference an earlier session, to continue interrupted sessions, to navigate between different concurrent sessions, to review the temporal sequence of sessions, and to substitute URL input or link following. All but the last one substituted missing session support. They further illustrate: “A single bookmark may represent an individual (or discrete group of) browsing sessions [...] One user wrote that bookmarks ‘take me to the last site where I was not finished during the last session.’ This user [...] used bookmarks as an inter-session history mechanism” (Abrams et al. 1998: 42f).

A major shortcoming related to session support is that a bookmark references only a single page. In addition, setting and organizing bookmarks is expensive – people are not willing to set several references actively. Usually they create only one bookmark per website or session (Abrams et al. 1998), even if they do explore – in the Abrams study in 44% of all cases – more pages than only the bookmarked one (Abrams 1997, page 3.12). Thus, it is only a first step in the right direction that the latest browser versions offer to save several opened tabs as bookmarks in a user-defined folder. Even this support lacks important history information, as only the currently loaded page of each tab is stored and not the tab-related histories.

Furthermore, revisits between parallel sessions are particularly poorly supported. Multiple windows and tabs for separating parallel paths provide only minor help. Revisitation becomes cumbersome as soon as the user wants to revisit a page in another concurrently opened window or tab. The Windows task bar provides buttons with pulldown menus that allow for navigation, however, they provide only very short and thus difficult to interpret textual abbreviations, and only list the currently frontmost tab per window. No information is available about the other tabs of the window or even the path of each tab or window. The user easily gets lost in-between in this complex environment of windows, tabs, and their individual histories.

Session-based history support – both for concurrent and for subsequent or resumed sessions – seems to provide large potential for improvement. Further research is necessary to determine how much revisitation support is beneficial, and what kind is appropriate. History tools that reflect the semantic entity of sessions and that enable the user to resume sessions and to revisit whatever page of the current session should be a reasonable basis for future improvements. The Session-Graphs approach aims at providing such a tool.

2.3.1.3 History Lists

Most browsers also provide global history lists (see e.g. Figure 2-8) that keep record of all visited pages across all browser windows. Web pages are represented textually,

in sequence or flat hierarchy, depending on the selected view. Potentially, any web page visited earlier could be revisited using these lists. However, the entries expire after several days or weeks and “very long-term revisits” are not supported.

Another drawback is that these lists are hidden in the depths of the browser’s interface. They are used very rarely, and some users do not even know about them. In the Tauscher study the history list accounted for only 0.65% of all browser interactions (1997a).

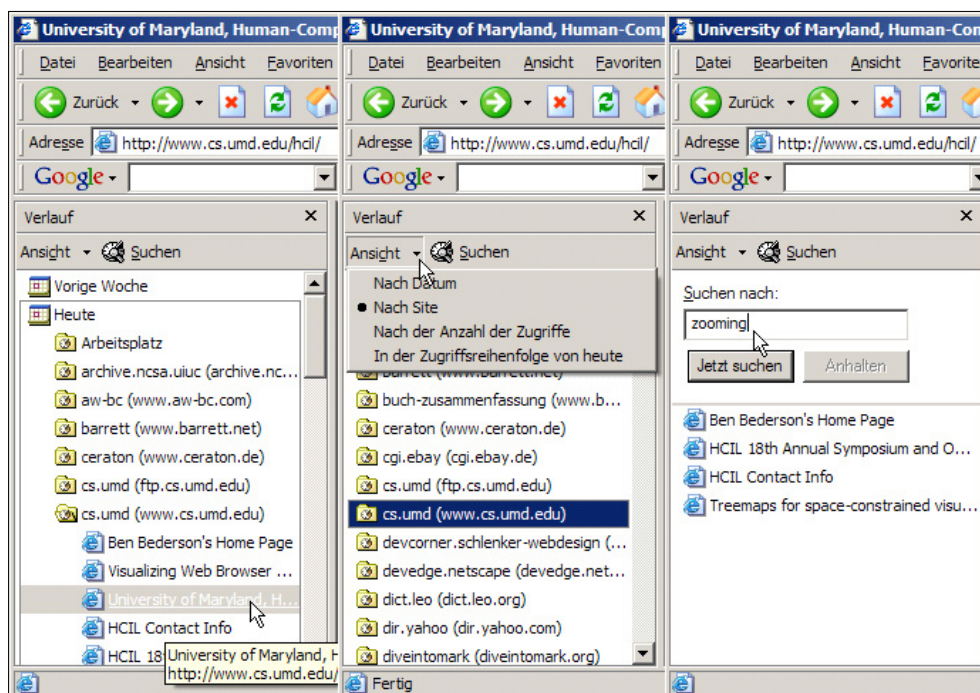


Figure 2-8: History list of Microsoft Internet Explorer 6.0 (3 views).

Left: sorted by day or week and clustered by domain; middle: sorted by site; right: text search in history.

In addition, finding a desired item in such a list is often demanding. Most history lists provide different methods for access. Internet Explorer 6 allows them e.g. to be grouped according to different criteria and to be searched textually. Four ways of grouping are provided – *by date* (by date of the last visit, e.g. today, Tuesday, Monday, previous week); *by site* (alphabetically listed domains); *by frequency*; and, *by today’s access sequence* (compare Figure 2-8). These choices seem to be rather technically motivated, based on what can be easily accomplished with the data at hand, rather than user oriented. Items can be simply sorted by frequency, however, that seems to be of little help. Users will have a hard time estimating the position of a desired item in a list that is sorted by frequency. Furthermore, after several days it is likely that a user can no longer remember the exact date when a certain visit occurred and the search function is limited to page titles.

An alternative approach that presents the user with clusters of visited pages similar to the current one in a hierarchy, or a flat structure, is introduced by Rizzo et al. (Rizzo, Daniel, Matera, Albertario & Nibioli 2006; Daniel, Matera, Rizzo & Press 2006) in their xMem project. Both approaches, however, share the shortcomings that valuable information about paths, sessions, window or tab states, and the user's tasks is lost.¹ In addition, these history lists provide almost no visual cues that would structure the presented text-based displays. Combined with the fact that the entries are deleted after a certain time, the user is eventually left in uncertainty whether or not it is worth searching the list or not.

The SessionGraphs approach will use another strategy to structure the user's history. It is based on the assumption that a user is more likely to remember the actual task he or she was executing while visiting a certain page, rather than being able to guess the correct date or to estimate the correct cluster. Therefore, it provides the user with the ability to explicitly represent the tasks he or she is working on.

2.3.1.4 Further History Techniques in Current Browsers

Web browsers offer additional history tools that are not equally well examined empirically. The *URL input field* e.g. allows a user to manually enter the address of a desired web page. An "autocomplete function" lists earlier entries that start with the same prefix as the typed URL. Most common browsers offer a popup menu to the right of the address field that shows the last manually typed URLs.

Firefox "quicksearches" further allow defining abbreviations for frequently used search pages.² Advantages are the conceptual simplicity and the small screen space it requires. Drawbacks include the cognitive effort necessary to memorize URLs, the restriction to single-page revisits and the loss of earlier session contexts.

In most browsers, the "*home*"-function allows quick access to a user-definable default web page, which is also loaded on startup. The home button, however, is used very rarely (0.9% of interface events, compare Tauscher 1996a: 21). Common start documents are web search engines, blank documents, individually built HTML pages with useful links, corporations' entry pages and intranet portals. The value of this rarely up to date history technique is very limited since it only comprises revisits to a single specific web page. The toolbar is already a full-value, improved substitute.

¹ Netscape Navigator 4.7 records only URL, title, date of first and last visit, and the sum of visits. Single visits, their durations, keywords, or contexts of pages visited before and after a certain page are not always logged.

² For example, defining the quicksearch "g" for a Google search and "l" for a search in leo.dict, a query "g history tools" or "l suche" in the URL entry field will be answered with the Google search results list on "history tools" or the leo.dict translation page of "suche".

Breadcrumbs are visual markers of links to already visited pages that are available in most browsers.¹ Usually color coding is applied to outgoing link anchors to enable quick recognition of recently visited nodes already in a departure node. Mark Bernstein introduced them to assist the user in avoiding undesired revisitations and guide disoriented readers back to familiar territory close to their current position (Nielsen 2004) – they create a history-enriched environment (compare Hill & Hollan 1993). Bernstein suggests limiting the number of breadcrumbs in order to keep them meaningful. He sticks to his metaphor: “imaginary birds remove bread crumbs the reader leaves unvisited for more than thirty pages” (1988: 43). This, however, can often be a problem in itself at the same time. Ultimately, they are not suitable for long-term revisits when they can expire already after several days.² Similarly, many websites disable breadcrumbs by using incompatible HTML sources that impede the browser when it tries to apply the common color codes. This leaves an inconsistent overall behavior.

Summarizing history support in current browsers, a broad variety of different history tools is offered. They are good enough to serve many frequently occurring cases of revisitation. The two most important problems are, first, the broad variety – each tool only addresses a specific revisitation problem and uses a different kind of user interface. Therefore, the user is forced to become acquainted with this multitude of techniques and interfaces upfront; they must train themselves in using them and remember how to handle them whenever needed. Secondly, each technique also has its own shortcomings. None of the current tools supports the reloading of an entire session including its history; no tool reflects the status of concurrently opened windows and would support the path-oriented navigation between them; no tool provides for the resumption of interrupted sessions.³

One of the most severe shortcomings is that all presently available techniques rely on recall rather than recognition – none of the approaches uses the power of information visualization. Existing approaches that suggest such history visualizations will be considered next.

¹ This concept differs from *breadcrumb trails* (Bernard & Chaparro 2000): These support navigation by representing the shortest path from the site’s top level to the current page (e.g., *Home > Products > Clothes > Shoes > Sneakers*).

² Default values for user definable expiration values:
Firefox 1.5: 0 to 9999 days, default: 9, and on demand.
Microsoft Internet Explorer 6.0: 0 to 99 days, default: 20, and on demand.
Opera 9.0: on demand.

³ Opera and Firefox just started to offer the possibility to continue with the set of web pages that were opened in different tabs and windows during shut down. However, the entire earlier internal session history is lost.

2.3.2 History Visualization and Related Projects

Norman (1993: 43) elucidates the inestimable value of external aids, such as visualizations: “The power of the unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities. How have we increased memory, thought, and reasoning? By the invention of external aids: It is things that make us smart”. This chapter introduces the principles underlying visualization and presents existing visual web history approaches.

2.3.2.1 Applying Information Visualization

The creation of proper history visualizations can free cognitive capacities for actual tasks by externalizing meaningful information. The major benefit of computer-generated, interactive, visualizations is to utilize recognition and perception instead of recall and memory, to amplify human cognition (Card, Mackinlay & Shneiderman 1999: 6). If well-designed, they shift the navigational activities to perception, freeing the mind for more interesting and important things. In most cases, recognizing things is considered being easier than recalling (Anderson 1995; Dhamija & Perrig 2000).

Shneiderman points out that our habituation of reading linear, sequential text for over 3000 years is a considerable burden to consume non-linear hypertext effectively (1998). Thus, already early hypertexts provided spatial visualizations for navigation (e.g. Storyspace: Bolter 1991b; Joyce 1991)¹, and to overcome the problem of disorientation (Nielsen 1990a). The entire process of *creating* hypertext was considered a spatial activity. Bolter (1991a) named it *topographic writing*: “Electronic writing is both a visual and a verbal description. It is [...] writing with places, spatially realized topics”. Our language reveals much about the application of the spatial metaphor. We talk about moving *back* and *forth*, about *home* pages, *portals*, *visits*, *chat rooms*, *site maps*, *downloads*, *Uniform Resource Locators*, etc. The approach of spatial hypertext (Marshall et al. 1994) (Marshall & Shipman 1993) (Shipman & Marshall 2000) made even further use of this metaphor.

One advantage of spatial representations is that we are used to creating internal representations, models and hypotheses, based on the properties of the per-

¹ A thoughtful discussion of spatial concepts and the web can be found in Bertel (2001).

ceived physical space.¹ When we experience a new physical space we first recognize specific routes, then landmarks, and finally build “mental maps”, a form of survey knowledge, of the entire region (compare Lynch 1962; Siegel & White 1975). Virtually, we cannot evade the impressions of spatial phenomena (Freksa & Habel 1990: 3). Our ability to process spatial representations internally is a prerequisite for dealing with the external world. Research already explored many related areas – such as mental rotation (Shepard & Metzler 1971), mental maps (Kosslyn, Ball & Reiser 1978), our ability to estimate relative positions (Stevens & Coupe 1978), and distances or directions (Hirtle & Heidorn 1993).

Web visualizations support the metaphor of a non-linear “information space”, *hyperspace*.² They are a special form of *information visualization*, the computer-aided visualization of non-physical data (Card et al. 1999: 7). They enable the user to transfer spatial concepts from the physical domain to the non-physical domain of the web. Similar activities appear, for example, in physical wayfinding (traveling to a known target, to an unknown target, and exploration, see Allen 1999) and in virtual environments (primed, naive, and explorative search, see Darken & Sibert 1996b).

A first challenge of any web visualization is that it requires valuable screen real estate and that it may become cluttered when it grows too large (Conklin 1987). Furthermore, unjustified conclusions may be drawn since the web is not *physical* space. Stanton et al. (1994) already criticized the unquestioned assignment of the spatial metaphor, further elaborated by Bertel (2001: 95ff). According to the latter, too many unwanted and unsuitable properties of the physical space – such as direction and distance – are motivated by spatial representations of the web, potentially leading to wrong conclusions.

Still, spatial history visualizations show convincing benefits besides *freeing cognitive capacities*. First, they enable immediate *orientation in a larger area*. While a web page provides only access to the immediately adjacent nodes (the »outgoing link context of depth 1« according to Mayer 2000) a history visualization can make very distant or even unconnected nodes directly accessible.

Second, they are *permanent reminders of the current task*, which is supportive when work is characterized by many interruptions or parallel tasks (compare Miyata

¹ Freksa und Habel (1990b) distinguish four classes of spaces: (1) physical space – in which the phenomenons of our environment reside, (2) psychological spaces – created by our perceptual and cognitive processing of physical space, (3) mathematical spaces – abstract structures defined by elements and axioms, and (4) metaphorical spaces – created by transferring spatial aspects of physical space to non-spatial domains.

² More on the importance of metaphors as foundation for human activities can be found in (Lakoff & Johnson 1980), as mentioned by (Maglio & Matlock 1998). Spatial metaphors are widely applied – for example, in the communication about emotion, success, and strength (concepts of »up« and »down«, see Freksa & Habel 1990: 4f), or about age, temperature, energy, and frequencies (concepts of size and directedness).

& Norman 1986). If well-designed, they *hide history information* that is not relevant in the current context. They only show potentially supportive information for the task at hand.

The potential drawbacks of history visualizations should be carefully weighed against these advantages when designing such visualizations. The following overview of history visualization projects, grouped by the kind of structural representation used to map the user's path, only considers *web history* visualizations and not visualizations of static *website structures* or dynamic *website usage*, as for example presented by Dodge et al. (2001) and Chi (2002).

2.3.2.2 The Path as 2D Spanning Tree

WebMap - the First Visual Spanning Tree History

WebMap (Dömel 1994, see Figure 2-9), one of the first approaches of visual history, shows a vertical spanning tree with elliptical nodes representing the visited pages. Color-coding is used to distinguish different properties and states of the visualized pages.¹ Doubleclicking opens a page in the browser. Nodes are entitled using numbers allotted in sequence of access. Undirected edges represent movements of the user – not necessarily implemented links.² An additional list view shows the temporally ordered accesses represented by URL and title. This list is tightly coupled (visually synchronized) with the tree view. Maps can be saved for later use. The system was written in *incr Tcl* and used together with Mosaic and tkWWW.

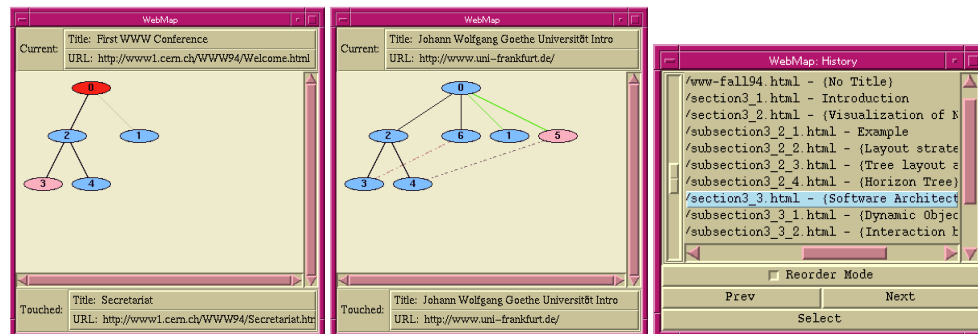


Figure 2-9: The WebMap history interface.

WebMap already contains the most important functions of a visual history system – visual, interactive representations, direct manipulation support, and spatial layout.

¹ Each node represents one web page. Color coding: normal node: blue; current node: red; mouse over: pink.

² Black: intra-site; green: inter-site; dashed: movements not part of spanning tree.

Shortcomings are primarily the difficulty in interpreting the number labels and the potentially misleading tree structure. It might be incorrectly mistaken as the internal hierarchy of a visited website – a problem that occurs whenever the spanning tree is taken to represent the navigation graph. It can be confusing when a site's home page is hidden deeply within the visualized tree, or when the root node of the visual tree refers to a page comes from the depth of a site. Further, the approach provides no concept of limiting a visualization's growth – it simply grows until the browser is closed. Thus, views may soon become large and cluttered. Session-internal navigation support is restricted to an ever-growing tree; session resumption is restricted to actively storing important visualizations as files for later use.

PadPrints - Spanning Tree with Thumbnails and Zooming

PadPrints (Hightower, Ring, Helfman, Bederson & Hollan 1998b; Hightower, Ring, Helfman, Bederson & Hollan 1998a, see Figure 2-10) shows a horizontal spanning tree using thumbnails of the web pages as nodes. New pages are added to the currently visited node as in WebMap. Labels have been improved by showing the first few words of a page's title.

The tree visualization is implemented using Pad++ (Bederson, Stead & Hollan 1994), which provides zooming functionality. Due to its multi-scale nature, each thumbnail can be smoothly enlarged up to the page's original size. A proxy-based design makes the approach independent of browser brand and version.

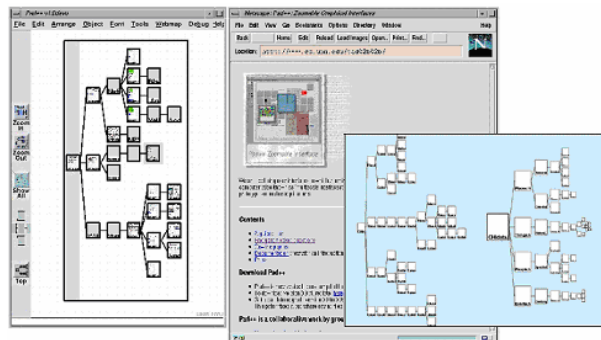


Figure 2-10: The PadPrints interface.

Two controlled experiments with 37 and 36 subjects showed that users were significantly faster and used noticeably fewer pages when revisiting web pages through the PadPrints tool compared to the plain Netscape Navigator 3.0. Furthermore, subjective ratings were significantly higher for the visual history tool.

PadPrints improved WebMap through thumbnails, zooming and the proxy. Still, the shortcomings are the potentially misleading tree representation and the inability to scale with growing histories, since only a single tree is provided. The

authors' suggestions for future research include the introduction of multiple trees, as well as the manual layout of the trees.

Domain Tree Browser - Separate Trees per Domain

The Domain Tree Browser project («DTB» – Gandhi, Kumar, Bederson & Shneiderman 2000, see Figure 2-11) improved several features of PadPrints. Gandhi et al. addressed the problem of quickly cluttered views by introducing separate vertical spanning trees per server-level domain and by a separate sequential list to select the domains. The resulting trees were smaller and grew at a slower pace.

The window is horizontally divided into three panes: The *domain pane*, the *tree pane*, and the *web page pane*. Pages are represented as thumbnails without titles. Color-coding is used to highlight the current domain and page. Thumbnail sizes reflect the number of visits to the related page. The trees can be zoomed and centered, and the history can be saved.

The prototype is implemented using a Java-based web browser (ICEsoft) and the Jazz Java graphics toolkit for zooming (Bederson & McAlister 1999; Bederson, Meyer & Good 2000), which is also used for the SessionGraphs approach. A study with four subjects showed improved times and lower numbers of visited pages compared to a single tree approach. The users, however, had difficulties in finding the desired domain in the list.



Figure 2-11: The Domain Tree Browser interface.

A commercial progression is TrailBlazer, a Mac OS X Panther web browser with visual history (MacWarriors 2004, see Figure 2-12). It was written in Objective-C using Apple's Web Kit Framework for rendering web pages.

The approach of separate trees per domain improves the scalability with growing histories over PadPrints. However, it introduces a new problem. As the user navigates between different domains, the first new page visited in this domain will be attached to the last page visited in this domain, no matter how long ago this visit

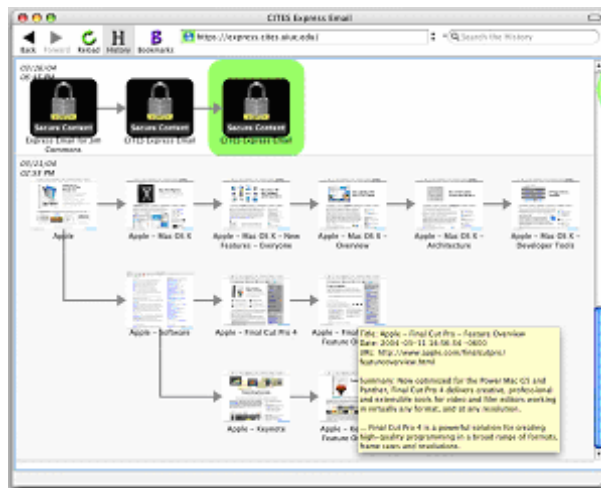


Figure 2-12: The TrailBlazer interface - an advancement of DTB.

occurred and whether there is any meaningful relationship between both pages or not. This behavior leads to confusing and semantically misleading visualizations. Furthermore, on the within-site-level the problem of cluttered trees still occurs when a user visits many pages in one domain over time. No further subdividing of trees is provided.

When two independent sessions are executed on the same server, they will be merged into the same tree visualization leaving the user unable to separate the different visits later. This will lead to problems comprehending the visualization.

2.3.2.3 The Path as 2D Graph

A few projects visualize not only the path's spanning tree, but the entire graph. This reduces the potential of misleading interpretations for the hierarchies used in path tree visualizations. This technique will also be applied by the SessionGraphs approach.

Nestor Navigator - 2D Maps of the User's Path

As soon as the user starts to browse, Nestor Navigator (Eklund, Sawers & Zeiliger 1999, see Figure 2-13) visualizes the path. As long as the user does not backtrack, the graph is drawn as a straight line of connected nodes. When the user revisits one of the nodes and browses into a new direction, the graph branches, resulting in a 2D graph. When the user navigates in a circle, the graph also shows the circle.

Beyond the plain history function, Nestor enables the user to structure and annotate visited nodes. Its authors distinguish “maps” - the complete graphs - and “bags”, which hold, save and share chunks of web pages. Colors can be used to encode specific sub-paths. Nestor is using the MS Internet Explorer engine, replacing its visual interface.

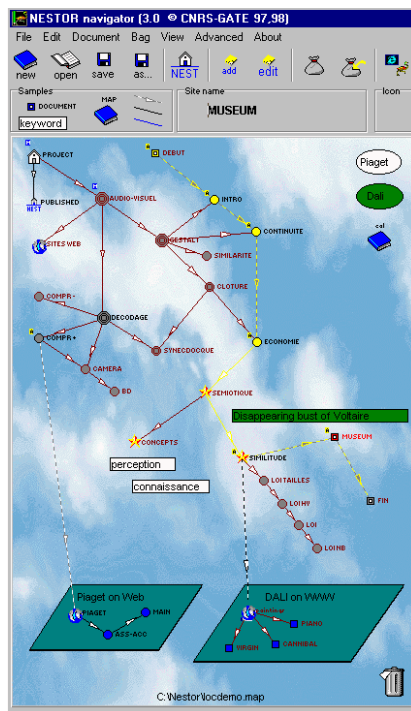


Figure 2-13: The Nestor Navigator interface - the path as a graph.

Nestor provides no concept and no interface for organizing the created maps. They have to be managed on the file level. In addition, explicit session and task support is missing. Finding a specific map or node at a later time period may become difficult as the archive grows. The path is visualized at a fixed scale – growing paths may soon reach the view’s border. It is conceptually designed as a visualization that is rather large – too large to be permanently placed alongside the browser on most screen sizes. The visual layout is not designed to save space. The user has to choose whether to view the history or the web browser window at any point in time. Then, reorientation may be difficult when switching back to the history.

WebNet

WebNet by Cockburn and Jones (1996, see Figure 2-14) displays the user’s path as a graph on a regular grid. Node titles are shown inside circles connected by directed edges. Bold outlines highlight the current page. Small dots represent the number of outgoing links. Node sizes can represent frequency, distance, or recency according to preselected settings. Two scales are available for viewing the graph.

Using the middle mouse button allows titles of pages connected by incoming and outgoing links to be shown. Web-subspaces are managed according to different starting pages of paths. WebNet is implemented using Tcl/Tk under the X Window system and runs aside the tkWWW web browser.

The WebNet visualization shows more details than other history visualizations (direction of edges, degree of outgoing links). A disadvantage is the regular

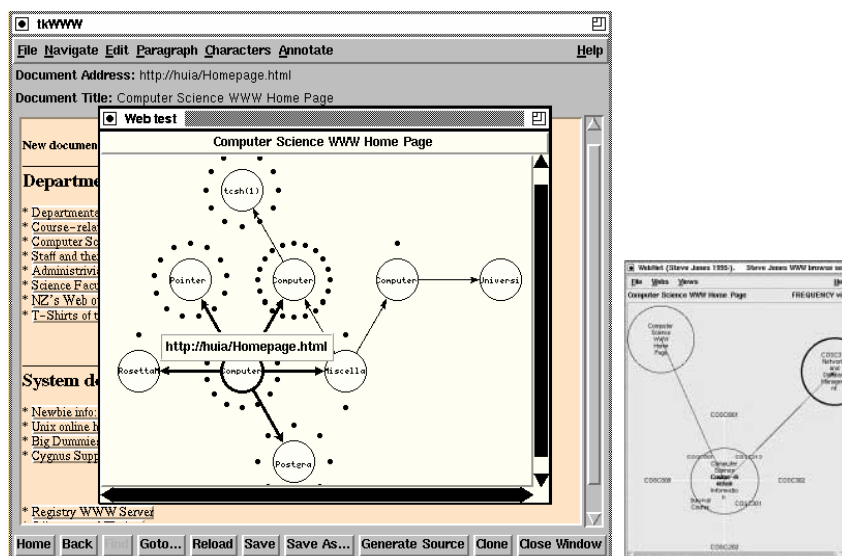


Figure 2-14: The WebNet interface - the path on a regular grid.

grid which at most allows six direct neighbors of a node. The authors do not elaborate on how a larger number of neighbors would be handled. Displaying titles within the circles makes the circles unnecessarily large. And, when many nodes are visited, the filled grid may be too regular to provide easily identifiable visual patterns and landmark nodes. Finally, neither session nor tasks support is provided.

2.3.2.4 The Path as 1D List

Several projects show the user's path, not as a spatial construct of visited and revisited pages, but as a temporal, one-dimensional list.

WebView & the GroupLab System

The WebView system (Greenberg & Cockburn 1999; Cockburn et al. 2003, see Figure 2-15) provides two automatically generated views, one of which is the linear, temporal view, with duplicates removed. The second view, the so-called hub-and-spoke-view, shows a hierarchy defined by the retrieved storage structure of the visited websites. Both views represent web pages as thumbnails that grow on mouse interaction. The thumbnails have been designed as history-enriched objects (Hill & Hollan 1993); a small dogear becomes darker as the number of visits increases. Explicitly set red dogears can be compared to bookmarks.

All thumbnails provide a popup menu with a list of page titles that can be accessed from this page. An evaluation with seven subjects showed significantly better revisitation times compared to Netscape.

The main advantages of WebView are that it is easy to understand and allows accessing the last few items easily. It needs no manual organization. The disadvantage, however, is that there is no additional structure, except the sequence of ac-

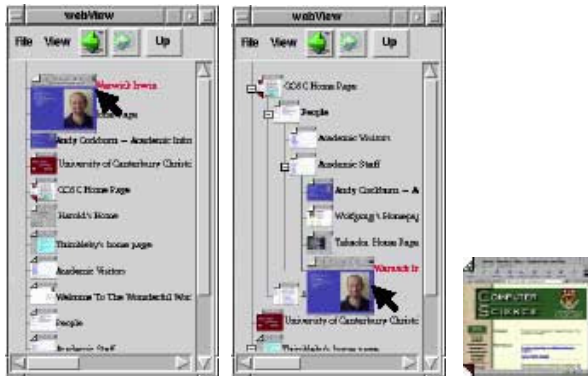


Figure 2-15: The WebView interface.
The sequential and hierarchical view and a thumbnail with dogears.

cesses. It could be difficult to recover a thumbnail in this list after weeks, even if it was bookmarked. The authors do not explain how long-term revisits are supported.

The successor of WebView, the GroupLab History System (Kaasten & Greenberg 2001, see Figure 2-16), removes the hierarchical view and adds dynamic query options (Ahlberg, Williamson & Shneiderman 1992) to the recency list (Kaasten & Greenberg 2001). This enables users to easily view pages with a high visit frequency or to view explicitly bookmarked ones. Nonetheless, there is still no structure within the bookmarks, which makes it difficult to navigate large histories.

The system is fully integrated into Microsoft Internet Explorer using MS Visual C++. It provides special back and forward buttons which allow navigating in the recency list rather than in the default stack-based history. The frequency of visits is mapped on the height of a small rectangle next to the thumbnail. A search field was added as well as a slider for dynamically filtering items. Moving the slider half way to the right, shows pages with higher visit frequencies; moving it further shows only explicitly bookmarked pages. This technique should allow faster navigation within the list (Ahlberg et al. 1992). Unfortunately, it still seems to be hard to predict which position of the slider produces which resulting subset of page representations. Navigating this history, therefore, seems to be rather difficult.

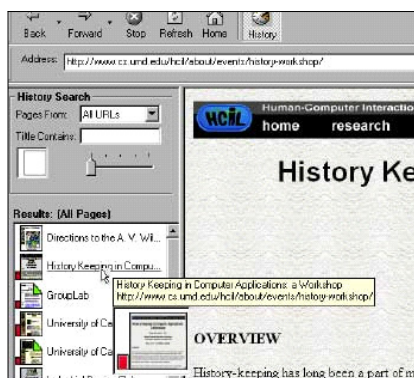


Figure 2-16: The GroupLab interface - an advancement of WebView.

Both of these systems only show the entire history at one specific point in time, or else a filtered subset of the history. There is no possibility to view session or task contexts. It relies on the user's abilities to successfully query and filter the list and to estimate how to navigate a filtered result.

2.3.2.5 Further Web Visualizations

Further approaches have been developed to visually support navigation on the web. They comprise pseudo 3D visualizations, metaphors borrowed from other domains such as books and libraries, and arrangements of items on a plane. Other approaches address the support of workgroups.

VISVIP – 3D Path Mapped onto 2D Site Structure

The VISVIP tool (Cugini & Scholtz 1999, see Figure 2-17) maps the paths of several visitors of a website onto a 2D visualization of the website's structure. The third dimension is used to map time information onto the path. It is primarily designed for website developers and usability engineers to analyze the paths of single users and to improve the site's structure and content. Its path mapping properties, however, could be easily transferred to a single user history scenario.

VISVIP maps the path as 3D spline onto a 2D representation of the website's structure defined by nodes and links. The 2D structure is laid out using a force-directed algorithm.

This approach is constrained to single sites. Initial experiments showed that using all three dimensions for the path results in visualizations which are difficult to interpret. Thus, only temporal information was mapped on the third dimension

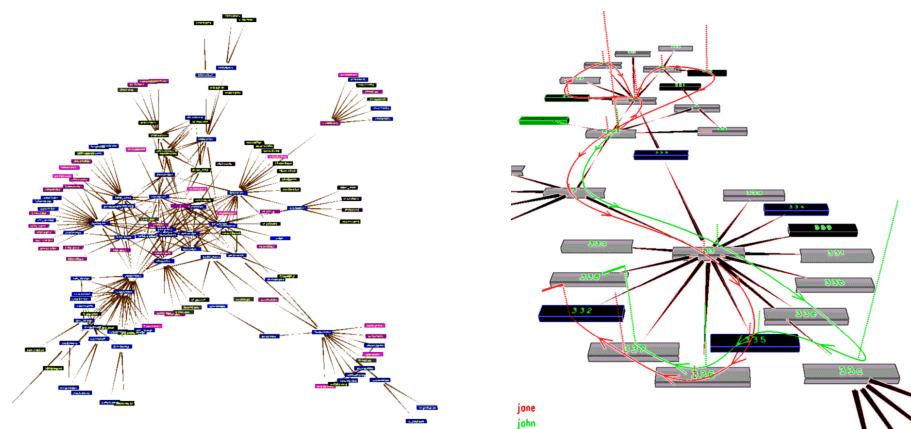


Figure 2-17: The VISVIP interface - website and user path.

Left: the site laid out using a force-directed algorithm. Right: a 3D path mapped onto the site's 2D representation.

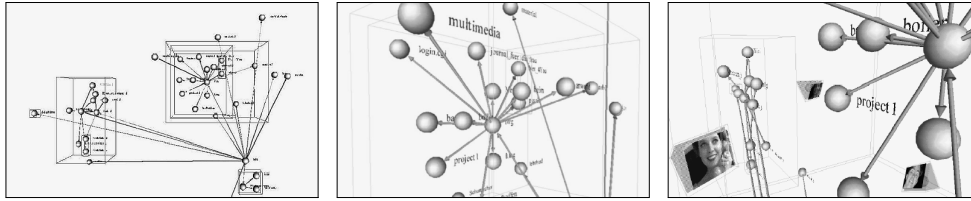


Figure 2-18: The CVP interface.

Left & middle: A part of a web server's content mapped onto 3D space.

Right: Avatars of three users represented near the documents they currently visit.

(staytimes per page). In contrast to the previous approaches, VISVIP not only maps the path, but also the site structure including unvisited pages. This is helpful for site designers and usability experts, and could be an interesting alternative for a web user to access their own history and further not yet visited pages of the site.

CVP—Mapping the Path onto a 3D Site Structure

In the Cyberspace Visualization Project (CVP – Adler, Heise, Mayer & Schefe 1998a, 1998b) we represented the hierarchical directory structure of websites as transparent nested 3D cubes (Figure 2-18). Each cube denotes one directory. Web pages in a directory are then delineated as colored spheres in the referring cube, and outgoing links as colored arrows. This approach is restricted to servers which organize their material in a meaningful structure on a file level – given in considerably many cases. We further provided the ability to visualize users that are currently online – and to represent them as small pyramid-shaped avatars next to the web page they are currently visiting. The purpose being to support communication amongst a site's visitors – a form of social navigation (Höök, Munro & Benyon 2003; Dieberger 1997). In a prototypical implementation, we successfully tested the system with five users.

The prototypical setup convinced us that it is a worthwhile undertaking to represent not only web-based information, but also its users as a means for social navigation and communication. History support, however, was constrained to the ongoing session.

WebBook and Web Forager—Applying the Book Metaphor

A WebBook (Card, Robertson & York 1996; WebBook_Video 1996, see Figure 2-19) allows to collect important web pages in a book-like way. The user can flip through the pages of a 2D projection of a 3D book in order to gain an overview. Spe-

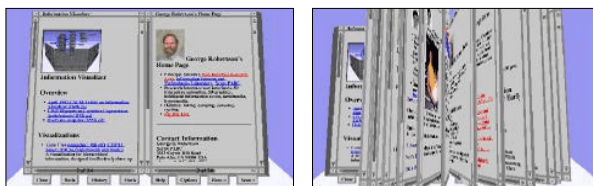


Figure 2-19: The WebBook interface.

Web pages organized in sequential order. Right: Skimming supports search and overview.

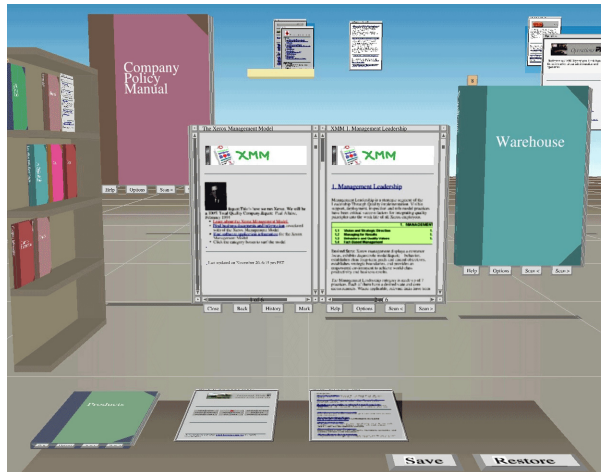


Figure 2-20: The Web Forager interface - organizing several WebBooks.
 Three layers of detail allow organizing and accessing a personal information space.

cial items can be bookmarked; collections can be defined by authors and readers.

Based on the book metaphor, the authors created an entire information workspace – the Web Forager – that allows a user to handle large collections of WebBooks effectively (see Figure 2-20). The workspace is divided into three levels of different detail. The front most “Focus Place” shows one item at a time in full detail; the “Immediate Memory” space in the middle allows the placing of items that are still in use, but not currently viewed; piles can be created there. The “Tertiary Place” allows the storage of several books in a shelf, which is shown at an angle to give hints about each book and to save screen pixels.

The book metaphor is easy to understand, easy to use, and it provides useful additional information, e.g. the thickness of a book, which itself gives a hint about the amount of content. However, the book-like sequentiality is a problem. It is structurally opposed to the non-linear structure of hypertext and problems occur when mapping a hyperspace onto this 2D space.

Data Mountain—Items on a plane in a 2D/3D approach

Data Mountain (Robertson, Czerwinski, Larson, Robbins, Thiel & van Dantzich 1998, see Figure 2-21) makes use of spatial memory for document management. Thumbnails of selected documents can be arranged arbitrarily on an inclined 2D plane in a 3D environment. They can then be manipulated using a simple 2D interaction technique.

This approach utilizes the human ability for remembering where someone has put something. To make the navigation through the items on the elevated plane easier, the user does not have to control a virtual camera. There is only one static point of view in the front of the plane. In order to navigate through piles of overlapping thumbnails, the metaphor of moving through high grass is used. The animation algorithm used by Data Mountain simulates single blades of grass that move



Figure 2-21: The Data Mountain interface.

It allows thumbnails of interesting web pages to be organized on an inclined surface in 3D space using 2D interaction.

temporarily aside and later return to their original position. A related 2D approach – named *Session Highlights* – is presented by Aula et al. (2005: 589) who display the outcome of a search as small thumbnails distributed over a 2D plane.

In Data Mountain, textures provide visual landmarks for distinguishing and memorizing positions and areas. It has been shown that this actually improves the readability and recognizability (Darken & Sibert 1996a). The Data Mountain showed significant improvement compared to Microsoft Internet Explorer 4 favorites in terms of access time and subjective rating (Robertson et al. 1998).

The spatial layout also showed good results for very long-term revisitation. The authors of a follow-up study report: “We were surprised to see how robust subjects’ memory for their layouts was” (Czerwinski, van Dantzich, Robertson & Hoffman 1999: 170). Users in this study, circa 6 months after creating their Data Mountains and 4 months after having last viewed them, were able to retrieve items at the same performance level, with only a brief retraining period. Interestingly, this ability did not depend on the presence of thumbnail images. A version of the prototype that showed white rectangles instead of thumbnails similarly showed good results (Czerwinski et al. 1999). Given that Data Mountain is not a fully immersive 3D application that is based on a static, immovable viewpoint in front of a 2D plane in combination with plain 2D interaction techniques, it encourages to assume that similar spatial placement techniques show comparable benefits when applied to pure 2D approaches, e.g. to *SessionGraphs*.

Drawbacks of the Data Mountain concept are, however, that the system does not handle the problems of scaling or systematic partitioning and that no session support is given.

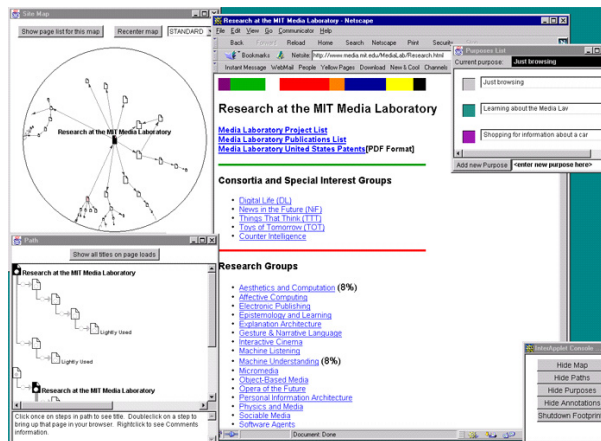


Figure 2-22: The Footprints interface.

The hyperbolic site map view (top left) shows the current document in the center and users' paths that cross this center. The paths view (bottom left) shows all single paths that contain the current document.

Footprints – a Group History Tool

Footprints (Wexelblat & Maes 1999, see Figure 2-22) provide the user of a website with information about the trajectories of other users on this site. The user can access this history via two visualizations: the site map and the paths view.

The *site map* window of Footprints is based on the technique of hyperbolic visualization (Lamping, Rao & Pirolli 1995). The center shows the currently viewed document, which is connected with other documents on the server that have been previously visited before or after the centered one by other users. Documents are represented as small rectangles with a red dogear and titles are shown in a separate window. Clicking a title highlights the corresponding node in the hyperbolic view; doubleclicking opens a document in the browser.

The *paths view* window shows single paths of other users in a tree view, where each path that contains the currently visited document is shown separately. The thickness of the lines represents the frequency of path use.

The consideration of other users' experience can be an advantage for users who are new to a specific site. They can use other users' paths as a guide to where they might find valuable information. A comparable project regarding the visual concept is WebQuilt (Hong & Landay 2001, without figure), which was designed to run web usability tests based on usage trace visualization for predefined tasks. Another visually similar concept is presented by Bertel (2001: 95ff). However, it only shows the user's own trajectories – all earlier paths that include the currently visited page. Bertel argues that his visualization prevents the *creation of* misleading spatial implications (2001: 124ff, compare Figure 2-23). Similar to the footprints map, the current node is static while the remaining visualization "moves underneath". Although this reduces the risk of wrong interpretations, this approach unnecessarily abandons half of the potentially interesting information – the nodes that precede

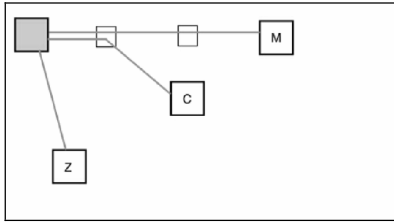


Figure 2-23: Bertel's sketch, combined paths comparable to Footprints.

the current one in earlier sessions. It also combines *all* previous sessions that contain this node, potentially mixing differently motivated sessions resulting in incomprehensible, difficult to utilize views. Since this project only existed as a concept, no study is available.

The most severe problem associated with both Footprints and Bertel's approach is that the view changes almost completely with every single navigation activity, since the paths that cross the current node will only be visible as long as this node is currently viewed. As soon as the next node is visited, all visualized paths will be immediately replaced. The user has to re-read the entire visualization to find orientation again and to make sense of it. This seems to be a profound disadvantage of both approaches.

2.3.2.6 Summary

The presented approaches applied different concepts of visualization to the domain of web history support. They differ in the data they choose to visualize and in the way they visualize these data. Several of these approaches proved in user tests that they could improve current history. They also offer valuable ideas to be considered in upcoming approaches. All the approaches, however, also show distinct shortcomings. In particular, none of the approaches presents a concise concept of supporting sessions (neither parallel sessions, interrupted sessions, nor similar sessions). Furthermore, none of the concepts offers mechanisms to organize the differently created views in a meaningful way (with the exception of the Web Finder, which supported the bookshelf and pile metaphor).

Most of the visualizations include no concepts for scaling with growing histories. Several tools are based on automated layout that does not allow manual reshaping to create more individual, meaningful and recognizable views. They also do not support "smooth animation" (except for Footprints) which would enable the user's eye to follow the modifications of the views more easily.

In many respects, Bush's metaphor is still applicable. The means we use to navigate the current web's highly advanced mazes are still the same as were used in the days of square-rigged ships.

2.3.3 Seven Critical Situations of Revisitation

As a result of analyzing the strengths and shortcomings of the aforementioned currently available or suggested history techniques, the following seven situations can be identified, in which proper history support is still missing. Future history systems should pay special attention to these situations.

Situation 1: Back to another Branch or Window - Revisit

Short-term revisits currently lack in two major areas, here subsumed as one critical situation. First, potentially valuable history information is deleted from the backtracking stack as soon as the user explores another path. Second, there is no support for the between-window or -tab navigation along the recent path. As long as these issues are not addressed properly, short-term revisitation remains in such situations rather arduous.

[Compare *Ariadne who cannot go back*, page 1]

Situation 2: Parallel Session Revisit

Current tools provide little means for separating parallel sessions and to navigate back and forth between their histories as desired. Going back to any earlier node of another parallel session should be easy. Currently, such situations are only manageable – if at all – by using multiple tabs or windows and navigating between them. This navigation, however, turns out to be cumbersome since task bar shortcuts and tab menu entries provide only little hints about the target pages and only reference the current page per window.

[Compare *Bacchus who is lost between his sessions*, page 2]

Situation 3: Session-Resumption Revisits

It has been shown that at least three – and often even all four – potential motivations for ending a session require its later resumption (compare page 28). However, no adequate support is available. Abrahms et al. found that bookmarks are often used as a workaround, even if highly unsatisfactory, since they only provide one entry point at a time and all earlier session-related path information is completely lost. Users, however, should be able to review all necessary session information easily and quickly, even after hours, days, or weeks.

[Compare *Chloris who has difficulties resuming her literature search*, page 2]

Situation 4: Rediscover Revisits

Revisiting a specific page, even after a long time, should be easier than visiting a page for the first time. Often detail pages are nested deeply in complex site hierarchies, not easy to navigate and hard to remember by address. Current auto-completion features, history lists, and breadcrumbs often expire too soon. In many cases, people use common, web-wide search engines for such long-term revisits instead. However, the actual finding of relevant information is often an iterative

process, especially for experienced users (Aula & Käki 2003). People are likely to encounter difficulties when they try to reformulate a query or to locate a search result within a complex results list (Wen 2003). Aula et al. (2005: 587) report: “Although the respondents seem to frequently rely on using a search engine to re-access material, this strategy is also problematic: *I think my main problem in web searches is nowadays that I can’t remember which were the terms that I used when I found a relevant site*”.

A further shortcoming of search engines as long-term revisitation tools is that the majority of web pages (the *deep web*, compare page 11) are not indexed by search engines at all. Estimations are that the most common web search engines index only one out of 3,000 pages (Bergman 2001). Thus, more than 99.9% of all web pages cannot be revisited using a search engine (ibid.).¹ Only history tools would be able to provide them again,² which is currently not an option.

[Compare *Daedalus who wants to review a certain picture*, page 2]

Situation 5: Same-Path Revisits

A common behavior is that users regularly revisit one distinct page by visiting some other starting page first and then following several links along a more or less familiar path. This wastes time and attention, and provides the risk for distraction. Such situations could be better supported by offering a shortcut to the desired target whenever the system detects that the user is about to navigate along such a path.

[Compare *Echo who is using the same paths over and over again*, page 2]

Situation 6: Similar-Task Revisits

Users should be supported in the execution of similar tasks – especially for the very long-term. For example, when a user once visited a certain country or a city and wants to revisit it now, then it should be easy for him to reaccess all the information he considers to be of interest (such as local information, hotels, travel information). Another example would be when someone has to create a report on a similar topic as before, it should be easy to rediscover interesting pages and supportive tools including online libraries, publication pages, and experts’ home pages.

Even if the web is becoming more dynamic, there is a good chance that important sources still exist after a considerable time has passed. Markwell (2003) found the half-life of a web page in the scientific education field to be 60 months. A clear sign that people want to revisit pages in the very long-term (see page 30) is reported by Abrams et al. (1998: 46). They found that users revisit almost all the

¹ Approaches for also opening up the deep web for search engines are only in their early stages (Huebsch, Hellerstein, Lanham, Loo, Shenker & Stoica 2003; Kabra et al. 2005).

² Revisits to dynamically generated web pages raise questions such as how to deal with reaccesses of already finished transactions. In such situations, it would be beneficial if the system were to allow only those actions that are reasonable by taking into account the later point in time. An in-depth discussion of these issues goes beyond the scope of this thesis.

bookmarks in their archives (96%) within the span of one year. The median, since the last visit, was about 100 days. Bookmarks, however, as shown above, are far from being ideal in supporting long-term revisits, and a tool that supports revisits to meaningful groups of pages is still missing.

[Compare *Phaeton who wants to execute a similar project half a year later*, page 3]

Situation 7: Group-Exchange Revisits

A last area of necessary improvement is group history support. In many situations it would be valuable to exchange history information with other team members. Although the result of a gathering task is currently only available for one person, shared histories could enable a more cooperative approach. This may include the exchange of sessions, but could also include entire tasks.

[Compare *Gaea who wants to inform her coworkers*, page 3]

The aforementioned seven situations illustrate those conditions, in which today's web users suffer most drastically by the lack of appropriate history support. Consequently, they will be addressed in the design of the SessionGraphs concept for revisitation, which is presented next.



3

The SessionGraphs Approach for Revisitation

Absent an explicit formal gathering interface, the main tool used in structuring the episode is simply the user's memory.
(Rosenberg 1996: 25)

This chapter presents the conceptual design and the prototypical realization of the SessionGraphs approach for revisitation. It is designed to overcome current shortcomings of history support by offering what Rosenberg demands, an explicit formal gathering interface based on the concepts of visualization, sessions, and tasks.

3.1 Problem and Goal

It has been shown that revisitations are essential for navigation on the web, especially in the context of information gathering (cf. chapter 2.2.3.2). They are vital elements of the hypertextual experience, both in the short and the long-term. However, current revisitation support still displays severe shortcomings (see chapter 2.3

and the summary below). The problem resulting from the diversity of highly specialized history tools is threefold.

Firstly, web users currently face a broad variety of different, self-contained history tools. They are burdened with learning the various interfaces, metaphors and ways of interaction, and have to be creative in finding feasible compensations for each tool's shortcomings. Thus, a first ingredient for the SessionGraphs approach will be to *provide a uniform, integrative history interface* and a metaphor that can be used for all kinds of revisits, both in the short and the long-term.

Secondly, current tools unnecessarily abstain from applying the benefits of visualization. They are mainly text or query-based and force the user to remember or query rather than to use perception and recognition. We find buttons and sequential textual lists instead of using easy to perceive history visualizations. Reducing the cognitive load by *applying appropriate visualizations* is the second ingredient.

Thirdly, current history tools widely ignore the user's working contexts of sessions and tasks. Revisits within the same session are poorly supported by the back button during explorative hub-and-spoke navigation, when using multiple windows, or when executing parallel sessions. The execution of several sessions simultaneously for one ultimate goal over long periods of time lack appropriate support. Bookmarks, URL entry and autocompletion all show severe shortcomings. Thus, the final ingredient are *sessions*, to represent single phases of subjectively coherent activities, *and tasks* to represent and organize activities with more superior goals that may comprise several sessions.

The goal of the SessionGraphs approach is to address these three aspects by combining the three mentioned ingredients: a consistent, integrative interface for web-based information gathering that makes use of history visualization (cf. chapter 2.3.2.1), and which considers the context of single sessions and the user's tasks (cf. chapter 2.2.3). The approach should be designed in particular to improve the seven situations of revisitation that require special attention (cf. chapter 2.3.3):

- 1) Back to another Branch or Window Revisits
- 2) Parallel Session Revisits
- 3) Session-Resumption Revisits
- 4) Rediscover Revisits
- 5) Same-Path Revisits
- 6) Similar-Task Revisits
- 7) Group-Exchange Revisits

Compared to current browsers, the goals are:

- 1) To reduce the *time* needed for a revisit.
- 2) To reduce the *number of unnecessary intermediary navigation pages*.
- 3) To increase the *subjective satisfaction* during revisitation.

Considered in a larger context, the developed approach addresses nine of the fifteen features that are missing in current web browsers, as already mentioned by Nielsen in 1995, which still were not even accomplished as late as 2005 (Nielsen 1995a; Nielsen 2005). The SessionGraphs approach either provides direct solutions to the above-mentioned features or supports a basis from which their accomplishment would be an easy next step. These features are listed below (*italic words indicate Nielsen's original demands*):

- *Overview diagram (global and local)* – SessionGraphs provides even more than that – visual overviews over all earlier site-spanning navigations.
- *Guided tours (paths)* – easy next step, see page 222.
- *Fat links (open many destination nodes at once)* – supported in principle by the session-based grouping. It would be a worthwhile next step to provide the possibility of loading all or selected pages at once, for example, into different browser tabs or into a specific window's path history.
- *Parameterized backtrack* – supported in principle by the session views that allow jumping directly to a certain node (type).
- *Visual cache* – this is exactly what the session view is; it permanently visualizes the last few visited nodes.
- *Flying through the information space* – easy next step, see page 163.
- *Link inheritance and clustering* – could be a valuable way to provide more scalable visualizations in the future, see page 221 (*scalability*).
- *Time-dependent notation (breadcrumbs do age in current systems)* – it would be easy to introduce aging in the session views in order to distinguish between nodes based on the time they were last visited. This could be accomplished, for example, by slowly blending the nodes color into the background color as time passes without a revisitation to the node.
- *Visual effects to emphasize navigational dimensions* – the session views provide a second person perspective to the individual history – in contrast, a first person view could visualize the user's path in a 3D world using a perspective as seen from the respective location on the path. The task chooser already provides an orthogonal perspective based on the user's task.

Providing a stable basis for introducing these long needed improvements should be a valuable contribution. Considering that revisitations are executed daily by millions

of web users on a global dimension, providing ideas for better supporting them should have a high and valuable impact on the quality and usability of the World Wide Web as a whole.

3.2 The SessionGraphs Concept

Thereafter, at any time, when one of these items is in view, the other can be instantly recalled merely by tapping a button below the corresponding code space. [...] It is exactly as though the physical items had been gathered together from widely separated sources and bound together to form a new book. It is more than this, for any item can be joined into numerous trails.

(Bush 1945: 45)

3.2.1 Supporting Sessions and Tasks

The SessionGraphs approach is designed to support the execution of subjectively coherent activities, both in the short-term and in the long-term, using the same visual metaphors and representations. To do so, the user can employ the construct of SG-sessions in order to represent intentional sessions externally. SG-sessions can be visualized, stored, used for navigation, and manually reshaped. The hierarchical structure of the user's tasks (according to the models of ART, cf. chapter 2.2.3.1) will also be visually represented. Consequently, a hierarchy of SG-tasks is offered for organizing SG-sessions.

Since manual classification requires serious cognitive effort (compare Jones, Munat, Bruce & Foxley 2005; Lansdale 1988), this approach adds additional organizational overhead to the user who has to explicitly name and organize the SG-tasks. However, this overhead is minimized as much as possible and the user is free to decide *when* to manage and (re-)organize his SG-tasks.

An SG-session automatically records and visualizes all the pages visited during a session. This is a significant modification which prevents the user from having to actively initiate the recording – as e.g. when using bookmarks. Lee (1992: 32f) coined the term *user collection tool* for such an approach, in contrast to a *selective log*.

The complete record offers the advantage that the user can directly revisit any node of an earlier SG-session, even if he or she did not originally deem this information to be of importance in the first place. Thus, this approach circumvents the “Delayed Value Judgment Problem” (Abrams 1997: 3-17) associated with traditional bookmarks. It further guarantees easier multi-page revisitation and session resumption even after long temporal absences.

The task concept within the SessionGraphs approach is comprised of a user-definable hierarchy of SG-tasks. The root node in this task tree is the SG-task “temp” under which all new SG-sessions are inserted automatically. The user can define new SG-tasks and sub-tasks also under this root node to create a meaningful structure for the SG-sessions. All the session-specific information, including visited nodes, paths and layout, is stored in the SG-session constructs. SG-tasks are purely containers for SG-sessions that organize them into a hierarchy. Inner nodes are always SG-task nodes and SG-session nodes are always leaf nodes.

The major advantage of the session- and task-based approach, in addition to the presentation of the user’s working context, is the fact that it systematically breaks down the history into meaningful, manageable units. This is currently not possible with existing approaches as presented in chapter 2.3, at least not in a systematic way that could be understood and applied by the user.

The SessionGraphs-user can freely navigate between all SG-tasks; one can resume earlier SG-sessions, switch between concurrent SG-sessions, and extend an SG-task no matter how old it is simply by adding new SG-sessions. One can also execute similar or repeated SG-tasks easily by using a copy of an SG-task. It is also possible to revisit groups of related pages easily by re-opening the earlier session visualization and reloading the pages of interest. An additional positive effect of the explicit task representation is that the user is reminded of his or her original task and may be less susceptible to distractions. Navigation along identical sub-paths is automatically detected and, as a shortcut, earlier SG-session visualizations are offered that contain the same partial path. Finally, SG-sessions can be exchanged with other people in order to establish a basic form of communication about web-related activities and / or to exchange interesting history information.

As previously mentioned, the SessionGraphs approach brings the additional overhead with it to maintain the task hierarchy and the SG-sessions. However, it remains up to the user to decide from case to case how much additional energy is spent on maintaining activities that can make future use easier. Furthermore, the user has exceptional freedom about the point in time for accomplishing this maintenance (i.e. before, during, or after session execution). If the visualizations are not maintained at all, new SG-sessions will remain in the default SG-task “temp”, which is the root node in the task tree. Session visualizations then will grow determined by the underlying layout algorithm (see pages 118ff), visually distinguishable from each other due to a random factor in the positioning algorithm and the different graph structures of the path. If the user actively maintains the visualizations, a new SG-session can be placed in any meaningful SG-task for later access and can be visually shaped in whatever way he or she prefers. It remains up to the user to decide to what extent the task and session structure should be maintained for later benefits. The results related to bookmark organization (see page 42) give reason to assume that many users are quite willing to spend some additional efforts for future benefits.

3.2.2 Applying Visualization

Both SG-sessions and SG-tasks are represented visually. A single visualization of an SG-session gives access to all the nodes visited during this individual session. A single visualization of an SG-task provides access to all the SG-sessions it contains.

3.2.2.1 Visualization of Sessions

In order to create appropriate session visualizations, the following principles have been applied:

- 1) Different SG-sessions should be easily distinguishable by their visual shape; each SG-session should have a characteristic, easily recognizable “Gestalt”¹.
- 2) Specific items within an SG-session should be easy to locate based on their visual appearance and their characteristic position within the graph.
- 3) The visualizations should mirror the “fluid” character of web-based activities – the smooth process of consecutive navigation actemes.
- 4) The visualizations should encourage the user to interact with them in an explorative, playful way.

Structure and Shape

A session visualization externalizes that part of the user’s path that belongs to a specific session. Structurally, this path can be interpreted either as a linear sequence of visits over time or as a directed graph of visited web pages (nodes) and moves between them (edges). The latter representation was chosen for the SessionGraphs approach, since a graph’s visualization contains more characteristic features than a plain sequential list (such as ramifications, loops, edge crossings, corner nodes, changes of direction, etc.). Skopik & Gutwin (2005) found that the memorability of nodes in a graph highly depends on such characteristic features, even when distorted by a fisheye viewing technique. An additional advantage of choosing the graph visualization was that fewer nodes need to be visualized (nodes that are visited several times are only visualized once), which saves valuable screenspace. This approach should ensure both that the nodes within a session visualization can be easily recog-

¹ The term “Gestalt” refers to the Gestalt laws, as further elaborated on page 76. The Merriam-Webster Dictionary gives a short definition: “a structure, configuration, or pattern of physical, biological, or psychological phenomena so integrated as to constitute a functional unit with properties not derivable by summation of its parts.”

nized based on their characteristic position, and also that different session visualizations can be distinguished from each other based on their overall shape.

A single web page in a session visualization should be represented as a circle, and – in the session view – with an attached title as a semantic label. Jones & Du-mais (1986) showed that the combination of semantic label plus spatial position as visual cues during document retrieval are superior to conditions that solely included a label or the spatial location.

The idea to visualize only the graph's spanning tree, as seen in the PadPrints and DTB projects (pages 50f), was rejected especially since the tree structure would not necessarily be intuitively associated with the traversed path and the hierarchy may be misinterpreted as the site's hierarchy. The path should instead be visualized as a directed graph with all its edges.

Hand-drawn sketches were used to find the desired appearance of SG-sessions. Examples of these sketches are shown in Figure 3-1. Figure 3-1, A–C, sketches three exemplary session visualizations. Their shapes are highly varied so that a user could easily identify one of them based on its shape. The left example may remind one of the letter “W” and the right one of “V”. Specific nodes could be identified as “the one at the crossing” or “the one at the nose” or “at the tail”.

Figures D to F show the growth of an SG-session visualization while the first few pages are visited. The growth should be dynamic and animated. Figures G to I show the desired behavior when a circle is closed by revisitation. After the circle is closed, as in Figure H, edge lengths are adjusted to a default length, in Figure I. Figure J shows a visualization enriched by specific encodings (color, size, shape) that can be used to communicate further information and to improve the orientation within the graph. Using these additional visual attributes should make it easier to relocate a certain node and to recognize the visualization as a whole.

Figures K and L show how detail-on-demand information is provided as the user hovers over a node with the mouse. A thumbnail, as well as further textual information is shown about the represented page.

One advantage of graph-based visualization is that human beings have been familiar with it for thousands of years. We are used to viewing, recognizing, reading, and distinguishing between graphs. Figure 3-2 shows examples of familiar contexts that use graph-based visualization. This example illustrates an important reason why nodes *and* edges are visualized in the SessionGraphs approach rather than only the nodes: it becomes much easier to locate specific items in such a visualization and to distinguish different visualizations that may seem similar.

Figure 3-2 (A–C) illustrates how stellar constellations already “discovered” in the night sky in ancient times made it easy to find orientation within the otherwise complex environment of single stars. The same effect can be observed in the children's game “connecting the dots” (E&F). It shows how easy it is to recognize shapes, once several nodes are connected by edges. Geographical maps (H&I) as

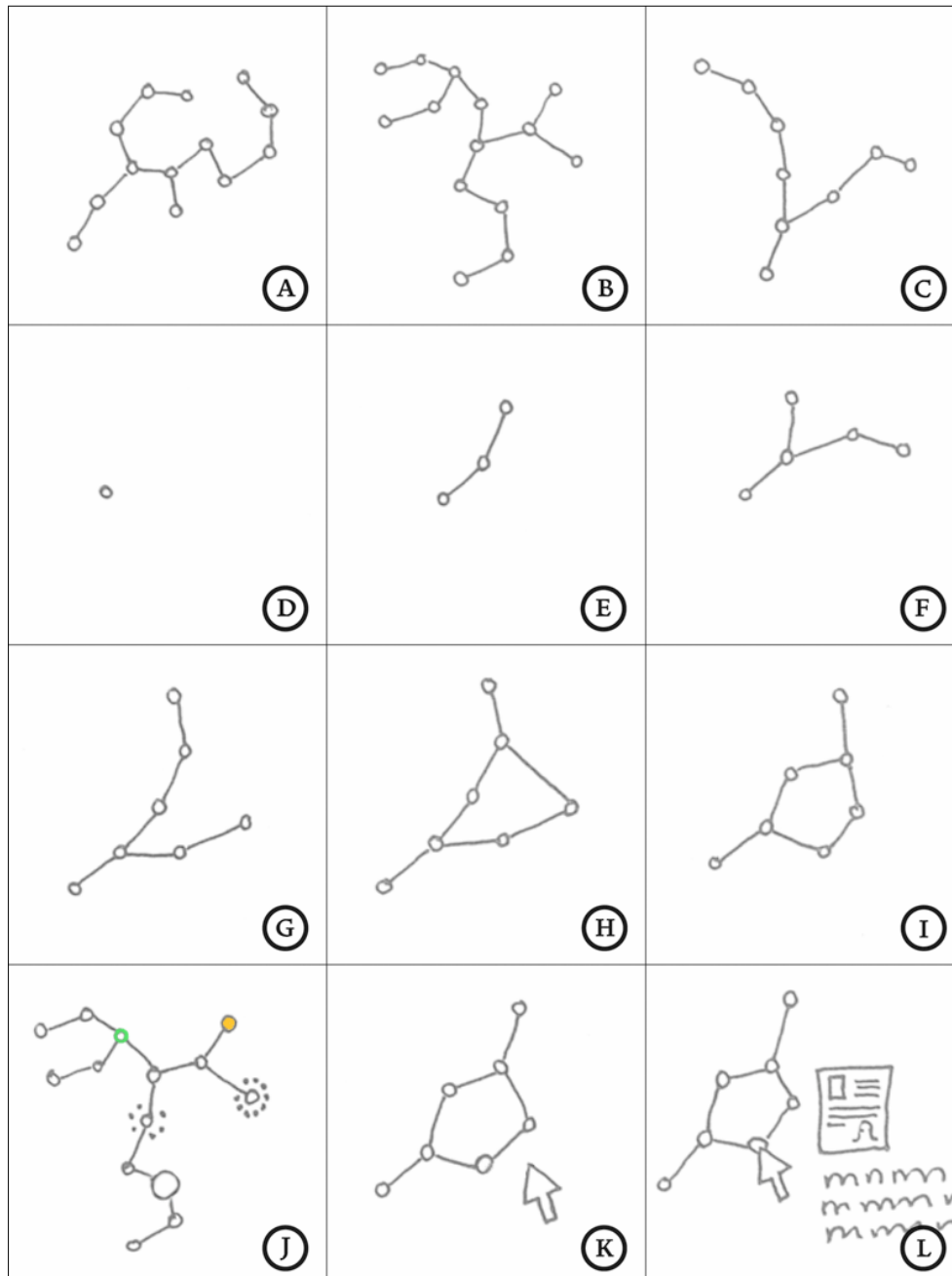


Figure 3-1: Early sketches of SessionGraphs visualizations.

A to C Three exemplary session visualizations.

D to F How one session visualization might grow.

G to I The user navigates in a circle, visually represented and fluidly animated.

J Examples for visual encodings of additional information: shape, size, color.

K & L Detail view is activated by hovering with the mouse over a node.

well as abstract ones (like mindmaps by T. Buzan, D&G) further illustrate the human ability to easily relocate specific positions within a graph-based visualization.

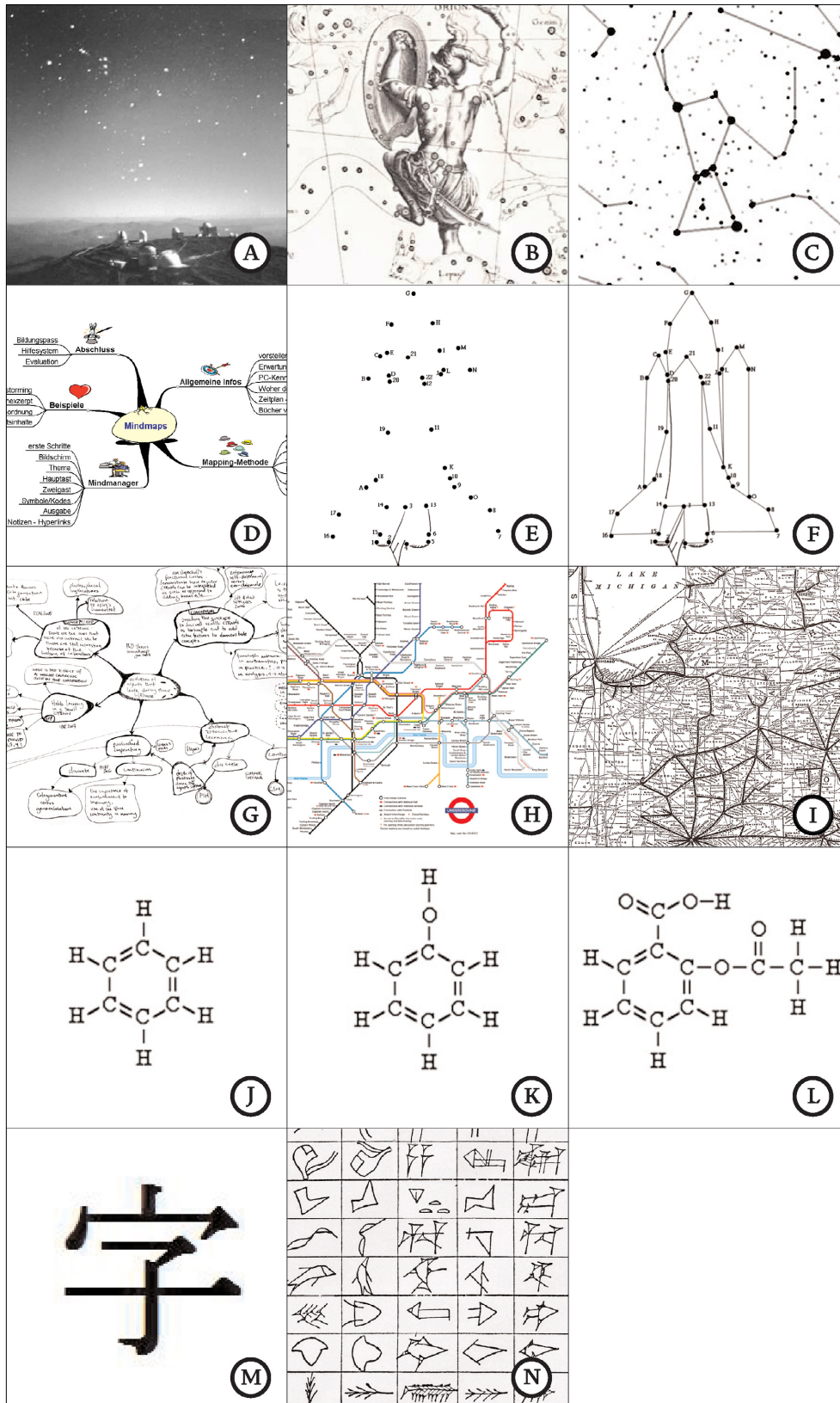


Figure 3-2: Cultural uses of the visualization of graphs.

A-C Stellar constellation "Orion"; D&G Mindmap; E&F Connect the dots (children's game); H&I Maps; J-L Structural formulae of chemical compounds (benzene, phenol, Aspirin®); M Chinese grapheme - ideographs "child" (子) and "roof" (宀) compound to create the meaning "to care for"; N Sumerian cuneiform script.

People usually become rapidly familiar with the metro maps of visited cities. After looking at them for several times, they know exactly where to find their home station or favorite destinations (“upper right, left to the big red node”, for example). Color codes, labels, edge crossings, and other graphical properties provide supportive landmarks for orientation. The structural formulae of chemical compounds (J-L) and the graphemes of Chinese and Sumerian script (M&N) demonstrate our ability to identify even the smallest visual modifications easily.

These examples vividly demonstrate the familiarity, advantages, and pervasiveness of graph-based visualizations. More convincing examples can be found in Tufte (1990: 45, 101-119). The presented advantages, wherever appropriate, are utilized by the visualization concept underlying the SessionGraphs approach.

Applied Visual Encodings and Gestalt Laws

Once the decision is made to provide a visual web history, it should be taken into consideration how the user could best receive the desired history information with the least amount of effort and distraction from the actual task. To achieve this, the SessionGraphs approach uses visual encodings and Gestalt laws. They allow shifting the communication of important information from cognition to perception. The underlying process of perceiving visual information can be divided into different stages (Ware 2000: 201f). In the early *feature extraction stages*, visual objects are analyzed according to *form, motion, spatial position, and color*. These properties are *pre-attentively processed* (Ware 2000: 163f); that is, they are processed in less than 10 milliseconds per item (Triesman & Gormican 1988). For this reason, they are used in the SessionGraphs interface to communicate important changes and new information. Motion and color, for example, are used to highlight newly added and currently visited nodes; small “satellites” around a node (form) communicate the number of outgoing links of the referenced web page, so that hub nodes can be easily identified (see Figure 3-1-J).

A valuable property of pre-attentive processing is that the time needed to identify encoded objects is independent of the number of surrounding “distractors” (Ware 2000). Thus, it can be used to communicate important information about single nodes “at a glance” – such nodes will “pop out” from their surroundings.

Table 3-1 presents the most important visual encodings applied in SessionGraphs.¹ The size of the nodes represents the accumulated stay time using a logarithmic scale, ranging between user-definable minimum and maximum values. The currently visited node is green; other nodes are grey by default. The user can color nodes manually to create recognizable landmarks. Each node is labeled with an automatically generated, editable abbreviation of the web page’s title. Edges can be

¹ They will not be addressed in the text in detail. To compare them with the actual interface please refer to Figure 3-4.

Encoded Information	Visual Encoding
New nodes SV, TV	Motion coding – new nodes gently move to their final destination starting at a more distant insertion point
Current page SV, TV	Color coding – green highlight
Hub pages SV, TV	Shape coding – nodes with more than 20 (40, 80) outgoing links are marked with 3 (6, 12) small round satellites
Number of visits to a page SV, TV	Size coding – nodes grow with the number of visits, based on a logarithmic scale
Search results SV, TV	Color coding – matching nodes have a pink outline, text attributes are temporarily set to bold style
Identical sub-paths SV, TV	Color coding – nodes of same sub-paths are colored light blue
Personal visual landmarks SV, TV	The user can manually color code important nodes in order to recognize them more easily and quickly
Current SG-session TC	Color and shape coding – blue rectangle behind textual title
Current SG-session TV	Shape coding – rectangular outline

Table 3-1: Visual encodings used to reduce cognitive load.

All listed encodings are preattentively processed.

SV: session view, TV: task view, TC: task chooser.

displayed with or without arrows indicating direction. Nodes and edges can be selected, moved, and deleted manually.

The second stage of visual perception is the *2D pattern perception stage*. Here, the overall shape as created by bordering contours can be recognized. The user perceives the shape of an entire SG-session – e.g. the “W” or “V”-like shapes above. In a third stage, structures between objects, in the present case nodes, are discovered using connection information, here, the edges. In the fourth stage the user matches the visualization with a stored representation, if it was seen before. This fourth stage becomes important when specific session visualizations are re-recognized after a certain time, based on their visual properties.

While the mentioned visual encodings address single visual entities, *Gestalt laws* (Ware 2000: 203ff; Koffka 1935) refer to the relation between several entities.¹ By applying them appropriately, the relation between several items can also be communicated on a perceptual rather than cognitive level. Table 3-2 shows how the

¹ Gestalt researchers asked themselves questions such as “What does it take to see a group?”, “How can 2D space be divided into perceptually distinct regions?”, and “When are two patterns recognized as being similar?”.

Gestalt Law	Application
Proximity	TV: To distinguish different SG-sessions. TC: To mark related SG-tasks and SG-sessions.
Similarity	SV & TV: To encode hub nodes visually similar. SV & TV: To perceptually group user-colored nodes.
Continuity	SV & TV: The chain of nodes along the path forms a naturally curved structure perceived as a continuous entity, supported by smooth and seamless transitions. Rectangular grids as used by e.g. PadPrints and DTB work against this law.
Connectedness	SV & TV: To mark the path.
Symmetry	Only modestly applied, since too much symmetry reduces orientational features. The layout algorithm creates exactly the desired degree of symmetry leaving irregularities for better orientation. Regular tree layouts (e.g. PadPrints, DTB) create more symmetry, which may make orientation more difficult.
Closure	TV: To mark the current SG-session by a bounding box.

Table 3-2: Gestalt laws used to support SessionGraphs visualizations.
SV: session view, TV: task view, TC: task chooser.

following *Gestalt laws* have been applied in the SessionGraphs approach: *proximity*, *similarity*, *continuity*, *connectedness*, *symmetry*, and *closure*.¹

The law of similarity, for instance, takes effect when users apply color codes to group similar nodes in a session visualization; identically colored nodes can be perceived at a glance. It should also be mentioned that Gestalt laws follow a certain hierarchy. The law of connectedness, for example, is stronger than proximity, color, size, and shape (Ware 2000: 207). Thus, connectedness was used for the user's path, the most important structure.

Summarizing, the applied visual encodings and Gestalt laws are utilized to reduce the cognitive load of using the designed visual history by shifting it to perceptual abilities. They allow the first two design guidelines to be obeyed – creation of a characteristic look on the level of a single session visualization and the ability to distinguish between several sessions.

Layout - The Fluid Surface Metaphor

To comply with the second pair of design guidelines – to mirror the fluid character of web activities and to invite the user to playful exploration – the *fluid surface metaphor* was introduced. The visual representations of web pages should slowly drift on a 2D surface, loosely connected by edges, representing the user's path. The behavior should be similar to clusters of sea roses or leaves that drift on a lake's gently mov-

¹ Connectedness was not an "original" Gestalt law (see Palmer & Rock 1994).

ing surface. As the metaphor suggests, the visualization should elicit a pleasant feeling, the feeling of an informal, ongoing activity, rather than the detached feeling often associated with the use of a formal, technical tool. The visual appearance should mirror the “fluid” experience that comes to mind by the widespread term “surfing the web” (Polly 1994, 1992). It should invite the user to manipulate the visualization, to manually reshape and investigate it playfully. The idea is that as soon as the position of one object is shifted, items that are connected follow this movement. The user should be encouraged to interact with the history, to use it, to design, to modify it.

Besides this intended metaphorical equivalence between visualization and activity, the smooth animation of the graph in the moment of structural change, e.g. when a new node is added or when nodes are removed or repositioned, is intended to support the user in maintaining object constancy to thus help him / her relate the two states of the system both before and after the change. Similar use of animation can be found in several earlier interactive systems, including the Information Visualizer (Card, Robertson & Mackinlay 1991), the WebBook and WebForager (see page 57), PadPrints (see page 50), and “Cone Trees” (Robertson, Mackinlay & Card 1991). Robertson et al. report on the design of these Cone Trees, which are hierarchies visualized as animated, pseudo-3D, cone-shaped trees that rotate on interaction: “Interactive animation is used to shift some of the user’s cognitive load to the human perceptual system. Consider what would happen if node selection displayed the rotated structure without animation. Since the rotations are complex, the user would take several seconds to re-assimilate the relationships between substructures. However, animation allows the perceptual system to track the rotation. The perceptual phenomenon of object constancy enables the user to track substructure relationships without thinking about it. When the animation is completed, no time is needed for re-assimilation” (Robertson et al. 1991: 190). In this sense, animation is also considered supportive in the SessionGraphs approach. Empirical evidence that animation supports users effectively in building mental maps from spatial information was discovered by Bederson et al. (1999), who “found that animation improves users’ ability to reconstruct the information space, with no penalty on task performance time” (Bederson & Boltman 1999: 28). Even if their setting cannot be directly transferred to the SessionGraphs approach, it still shows many parallels.

To design the visual behavior of SessionGraphs, the movement of items drifting on a water surface was examined in an informal setup (see Figure 3-3). A large Petri dish (\varnothing 30 cm) was filled with water. Subsequently, approximately 200 black, round paper pieces (\varnothing 0.5 cm) were carefully dropped onto the surface and filmed by a video camera. Several characteristic behaviors could be observed. Smooth movement started immediately from the moment the pieces were dropped into the water. Contiguous pieces were attracted to each other and formed clusters with several loosely moving “arms” – a behavior later reflected by the session visu-

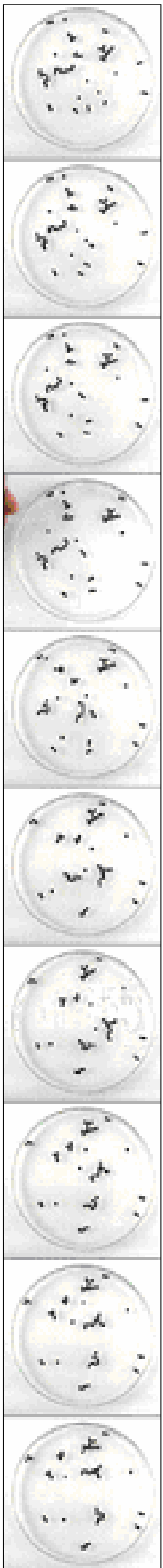


Figure 3-3: Animation algorithm derived by water motion observation.
Video stills taken from an informal setup to analyze the behavior of drifting and moving items on a water surface. Used to obtain properties of the animation algorithm.

alizations. When movement was enforced by slightly blowing air across the surface, the clusters repelled each other – later mirrored in the miniature graphs’ animation (s. page 78). The animation algorithm that was inspired by these observations is presented in chapter 3.3.2.2.

The fluid surface metaphor produces two welcome side effects. First, newly added nodes start to move immediately until they reach a stable position in relation to the other nodes. This movement is pre-attentively processed, giving the user an unobtrusive sign on where the current page is represented in the session view. Since this motion lasts for several seconds, it works even if the user’s initial focus was on the web browser window and not on the SessionGraphs’ GUI.

The second side effect is that it decreases the risk of misinterpreting the spatial layout. The chosen movement of nodes communicates, first, that the absolute position of a node is not meaningful, and second, that the relative position of two nodes that are not directly connected by edges is also not meaningful. This gives the user valuable hints about how to interpret the visualization.

Dynamic Visualization Behavior

The user can create new SG-sessions or switch between existing ones at any time. Session visualizations are created to comprise all the visits that cohere in the user’s mind, even if they span several server or window borders. In contrast to current back buttons that hold separate history stacks per window, this enables the user to return easily to pages visited in other browser windows. A new node is always added to the *current* SG-session – i.e. the session currently visualized in the session view. It is attached by an edge to the node that was last visited in this SG-session. The node is given a space-saving, brief title, created by the concatenation of the first word of the corresponding web page’s title with the first character of the second word of this title. In case a subsequently visited page produces the same title as a page before – as is likely to happen when several pages are visited on one server – then earlier used words are omitted by the algorithm in the ensuing titles. A web page with the title

“Tate Modern: International Modern and Contemporary Art” would be abbreviated to “Tate M.”, for example. A next page “Tate Modern: Current Exhibitions” would be abbreviated to “Current E.”, since both “Tate” and “Modern” were recently used and are thus omitted. Less meaningful words, e.g. articles, are omitted by an exclusion list. Thus, the titles “About Modern Art” or “The Modern Art Museum” would be abbreviated to “Modern A.”.

Nodes are inserted by a positioning algorithm that minimizes edge crossings. In the case where an already represented web page is revisited, no new node is inserted, but instead the existing page representation is color coded as the currently viewed page. Experiences gained during the explorative prototyping phase showed that users prefer that edges are inserted only when link are traversed by link following in the web browser, and not during after interaction with the SessionGraphs GUI. Incidentally, this also keeps the number of edges smaller and the view more comprehensible.

The visualization is animated as soon as a node is inserted, deleted, or moved. The underlying algorithm, derived from the water surface observation experiment, models a spring system between the nodes that will reach an energetically optimal state in which attractive and repelling forces between the nodes are balanced. This process of reaching the optimal state is spread over several seconds in order to keep the modification observable. Thus, while surfing, a “fluent” process can also be perceived in the visualization.

Formal Definitions

A *session visualization* (SV) is a pair of a *session graph* (SG) and a *sequential visits list* (VL): $SV = (SG, VL)$. The session graph determines the visual appearance of the session; the visits list enables chronological backtracking. A *session graph* $SG = (V, E)$ is a directed graph laid out on a 2D surface. It is a pair of V and E, where V is a set of vertices and E a set of edges (ordered pairs of distinct vertices) that are of the form (u, v) , such that $u, v \in V$. $|V| = n$ is finite and called the *order* of SG. It represents the number of different web pages visited in the respective session. Each vertex $v \in V$ represents exactly one web page. $|E|$ is called the *size* of the session graph.

The *visits list* $VL = \{List, BackPos\}$ maintains a temporally ordered list of all visits, $List = (vis_1, .., vis_n)$, made during a session, including all duplicates. A pure recency list would not work – since it oscillates between two nodes during consecutive backtracking. Instead, the integer value *BackPos* is used to store the current position within the variable *List*. Each *visit* $vis \in List$ is a pair of a web page and a time stamp that represents the start time of the specific visit.

Having clarified the nomenclature, the SessionGraphs interface including the visualizations is introduced next.

3.2.3 Providing an Integrative Interface that Supports Flow

The interface concept of SessionGraphs is to provide one single, consistent, uniform interface to support all types of revisits. Users only have to learn one visual metaphor, one set of visual codes, and one way of interaction. There will be no hidden history list that is concealed from the user. The user can be sure that any history item can be found in the logged history using this single interface. One of the participants in our web usage study reported that, in several situations, he was no longer sure whether he had bookmarked a certain important page, had used his scrapbook extension, or simply a local copy of the file to re-read it later. He was clearly in doubt where to search for it and certainly disrupted from his flow. The integrative interface should overcome these shortcomings of a highly heterogeneous, inconsistent history environment by applying a well-designed cost hierarchy. A general interface approach was chosen so that its users can easily learn to use the history functions but still keep their concentration on their actual work.

The same motivation is expressed by Benjamin Bederson who – more generally – demands to build “Interfaces for Staying in the Flow” (2004) – claiming that “Computers should help us concentrate on our work, without concentrating on the computer”. He refers to the psychological term *flow*, as coined by Csikszentmihalyi (1991), characterizing the “optimal experience” of concentrated, non-interrupted activity based on an appropriate balance of “challenge and skills” (citations from Bederson 2004: 1). Even if the SessionGraphs approach was designed before Bederson bridged Csikszentmihalyi’s work to interface design, his broader perspective on HCI can be applied profitably: “By focusing too closely on narrow quantitative measures, our field risks missing out on other important characteristics of what makes an interface ‘good’” (ibid.). It will be further elaborated below on how the SessionGraphs approach meets the requirements postulated by Bederson in order to maintain flow.

A Cost Hierarchy for History Access

Since any history visualization requires screen real estate, which is a very limited and valuable good, and since no history visualization is able to visualize the complete history all at once, the designer of a history visualization tool has to decide, which history items should be visible at a certain moment and how other items can be accessed. Some items will be most easily accessible (lower costs) while others require little and still others more effort (higher costs) to be retrieved from the history tool. This cost hierarchy has to be well-designed in order to make frequent and probable revisits easier than other revisits.

The SessionGraphs GUI is designed based on a three level cost hierarchy (see Table 3-3). Accesses within the same SG-session *and* SG-task (level I) are easiest, they can be executed just by clicking a node in the current SG-session. Level II is also

Level	Degree of difficulty	Context of revisit	SessionGraphs GUI actions necessary for revisitation
I	easiest	same SG-task, and same SG-session	CSV
II	very easy	same SG-task, but different SG-session	CTV, or SST → CSV
III	easy	different SG-task, and different SG-session	STT → CTV, or STT → SST → CSV

Table 3-3: Cost hierarchy of history access that considers the users' needs.

Abbreviations:

CSV - click a node in the current session view

SST - select desired sg-session either in task chooser or in task view

STT - select sg-task in task chooser

CTV - click a node directly in the task view (miniature graphs)

A → B - first do A, then B

defined by the same SG-task, but within a different SG-session. Here, the task view can be used to easily switch to another SG-session in the same SG-task. Finally, to revisit a node in another SG-task and SG-session (level III), the user can use the task chooser to select the appropriate SG-task and SG-session. Additional means are provided to access history items located in levels II and III more directly: a full text search allows finding any node, SG-session, or SG-task directly, based on its textual description; furthermore, tabs allow switching easily between the last three session visualizations, no matter what SG-tasks the SG-sessions belong to.

The SessionGraphs Graphical User Interface

Figure 3-4 shows the Graphical User Interface (GUI) of the implemented SessionGraphs prototype on the right side of a web browser's window. It is a standalone window – a narrow, vertical rectangle, manually adjustable in size. The user can minimize it to the task bar at any time. The GUI is optimized for screen sizes of at least 1024 x 768 pixels, although 1280 x 1024 pixels and more are preferred. The GUI can be used in two display-modes; one mode automatically aligns the browser windows to the left of the GUI and maximizes windows size. This guarantees that browser and SessionGraphs windows do not overlap. Alternatively, the SessionGraphs and browser windows can be positioned manually and independently.

The topmost elements of the GUI are the *menu bar* and the *toolbar* (A). The menu bar gives access to all functions of the system, including creating new SG-sessions or SG-tasks and accessing the preferences dialog. Table 3-4 presents all menu bar items. Keyboard shortcuts (shown in the right column) and/or mnemonics (underlined characters) are provided for the most important items. Frequently used functions are also accessible via the toolbar (see Figure 3-5). Table 3-5 explains the different toolbar buttons.

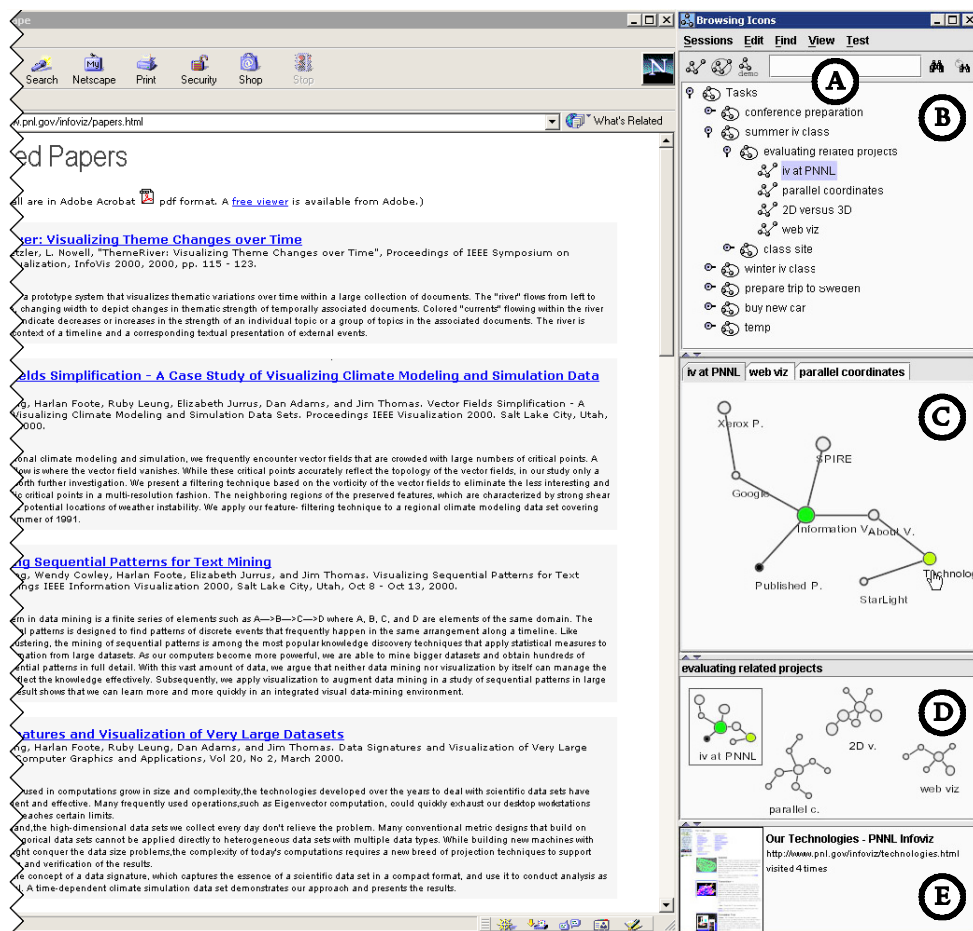




Figure 3-4: The SessionGraphs GUI to the right of a common browser.

A - menu bar and toolbar, B - task chooser, C - session view, D - task view, E - detail view.

Next to the *New Session*  and *New Task*  -buttons, two alternatives to the browser's back button are offered for browser backtracking, accessible using the two buttons with arrow-icons. The first one is a simple backtracking that loads the page at index $i-1$, where i represents the position of the currently displayed page in the *visits list's List*. When the button is clicked, the pointer *BackPos* is updated, the browser loads the page, and the visualization highlights the respective node. The second way is provided for backtracking after misleading navigation or undesired sub-paths. Here, the user can go back and simultaneously delete the currently visited node and adjacent edges from the visualization.

The *task chooser* (Figure 3-4-B and Figure 3-6) allows viewing, organizing, and accessing the complete history in the form of a hierarchy of SG-sessions and SG-tasks. An orthogonal 2D tree layout is used, well-known from common directory trees. It supports familiar interaction principles (cut, copy, paste; drag & drop). Alternative to browsing the outline view, the user can textually query the history for textual strings.

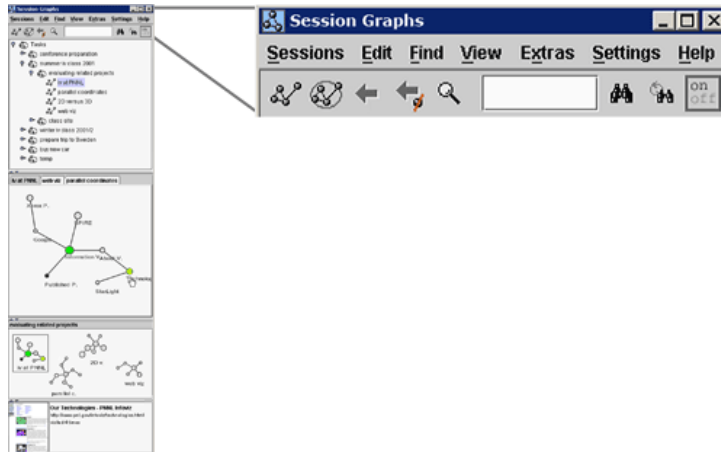


Figure 3-5: SessionGraphs GUI (1) - menu bar and toolbar.

Sessions Menu	Description	Short Cut
<u>N</u> ew Session	Creates a new session	CTRL-N
New <u>T</u> ask	Creates a new task	CTRL-T
<u>S</u> ave	Saves the current history manually. Can be used instead of waiting for autosave	CTRL-S
<u>E</u> xit	Ends the program	CTRL-Q
Edit Menu		
Go <u>b</u> ack to previous version	Undo - Navigates back within 20 internal states	CTRL-Z
Go to next version	Redo - Navigates forward within 20 internal states	CTRL-Y
<u>D</u> elete	Deletes the selected item(s)	DEL
Find Menu		
<u>F</u> ind	Searches the string provided in the adjacent field in titles, URLs, task and session labels. The result is shown in the detail view with highlighted query string.	CTRL-F
Find <u>n</u> ext	Shows next search result in the detail view	CTRL-G
View Menu		
<u>A</u> nimation On	Toggles the session graphs between animated and not animated	CTRL-A
Make current view liquid again	Resets a currently "frozen" session graph - the whole view is laid out again automatically using force directed placement	
<u>C</u> lose top Sessionview	Closes the frontmost session view. Activates the underlying	CTRL-W
<u>C</u> ollapse tree view	Closes all opened tasks and subtasks in the task chooser	
Extras Menu		
Save a comment	For long term study participants - to save comments on the tool's behavior (bug reports, suggestions for improvement) directly into the logfile at the current position.	
Check for new mailed sessions	Loads session files stored in a specific directory into the task chooser	
Mail session to a friend	Saves a file version of the current session into a predefined directory - in order to attach it to an e-mail	
SessionGraphs Version: 0.91	Displays the current system version	
Settings Menu		
<u>C</u> onfigure settings ...	Opens the "configure settings" dialog	
Help Menu		
SessionGraphs Online <u>H</u> elp	For study participants - Loads the SGA online help page into the browser	F1

Table 3-4: Menus and menu items, mnemonics (underlined), and shortcuts.









Toolbar Button	Description
	Begin a new session
	Create a new task
	Go back to previous node
	Delete current node and go back - alternative to browser back
	Search the web: starts new session with definable search engine
<input data-bbox="395 524 443 562" type="text"/>	Text entry field (for use with find and find next)
	Find in locally stored history (titles, URLs, session & task labels)
	Find next: search results are presented in detail view
	Toggle system between logging (ON) and not logging (OFF) mode

Table 3-5: Toolbar items and their functions.
All buttons are supported with tooltip explanations.

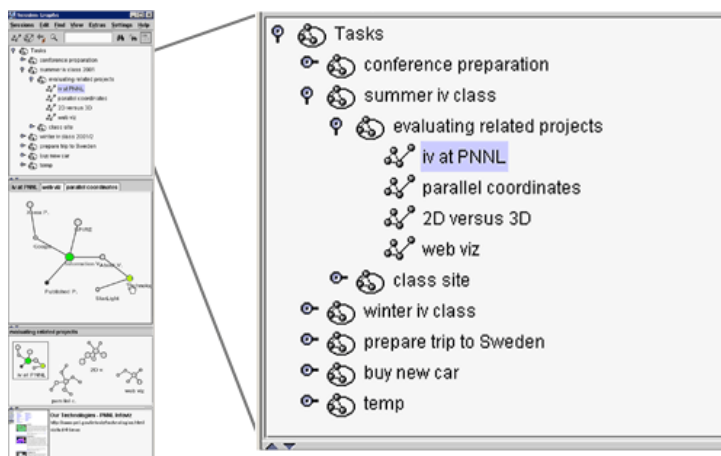


Figure 3-6: SessionGraphs GUI (2) - the task chooser.
Allows accessing and organizing the hierarchy of SG-tasks and SG-sessions.

The *session view* (Figure 3-4-C and Figure 3-7) shows the visualization of the current SG-session in full detail. It provides the fluidly animated, interactive view that allows the direct revisitation of any related web page. In case that a session graph starts to exceed the rectangular output area, at first the entire visualization is shown at a smaller scale (factor 0.8 instead of 1.0) so that more items fit into the rectangle. When the graph exceeds the rectangle also at this scale, the visualization starts to scroll. The visualization's layout can be manually modified; nodes can be selected, edited, moved, and deleted. Hovering over a node or its title brings up more information in the detail view. Clicking loads the page in the browser. Right-click opens a popup-menu aside the selected node, which allows to directly modify-

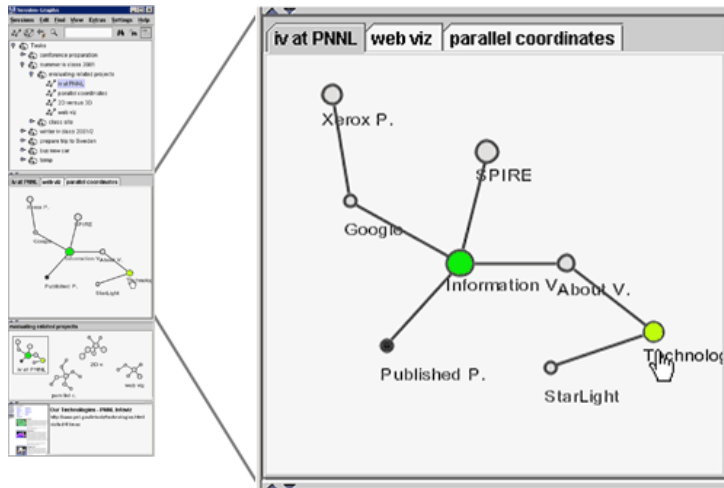


Figure 3-7: SessionGraphs GUI (3) - the session view. Provides views of the last three SG-sessions on different tabs.

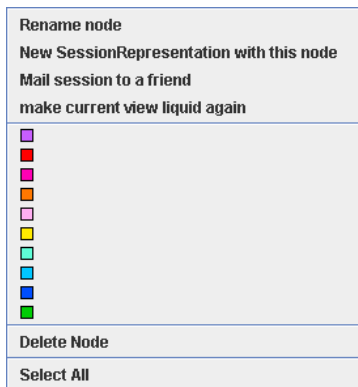


Figure 3-8: Popup-menu for node modification in session- & task-views.

ing the node accordingly (see Figure 3-8). The user can manually create new SG-sessions, delete old ones, rename them, move them, and copy them. Up to two additional tabs provide access to the views of the two SG-sessions that were shown before.

At any time, exactly one SG-session is the “current session”. It is displayed in the frontmost tab and highlighted both in task chooser and task view. All subsequently visited nodes are then added to this session. When the user selects an SG-session from any of the views, the selected SG-session then becomes the current one. When the user selects an SG-task, the first SG-session that is found either in this SG-task or in one of its sub-tasks will be the current SG-session from then on.

The *task view* (Figure 3-4-D and Figure 3-9) presents all SG-sessions that are direct siblings to the current SG-task as interactive *miniature graphs*. Clicking them opens the related session visualization in a session view tab. Clicking a node in a miniature graph directly loads the referenced page into the browser and opens the

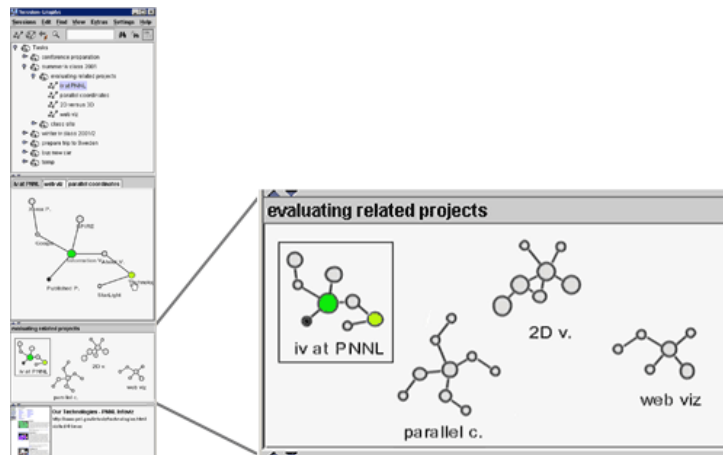


Figure 3-9: SessionGraphs GUI (4) - the task view.

Shows miniature graphs of all SG-sessions that are direct children of the currently selected SG-task. The rectangle identifies the current SG-session. Orange nodes are Hera's color coded publication pages (s. page 22).

related session visualization as well. Miniature graphs contain less visual detail compared to session graphs (missing page titles), but show exactly the same visual shape. Single nodes can be identified either by recognizing them according to their spatial position, graphical attributes (color, size, highlight, satellites), or by hovering with the mouse over them to show further detail in the detail view. Task view and session view are tightly coupled¹ so that hovering over a node or dragging a node immediately influences both views similarly. Miniature graphs can be manually arranged on the 2D plane. Overlapping is prevented by the layout algorithm. The map is smoothly and dynamically animated after modifications.

The *detail view* (Figure 3-4-E and Figure 3-10) provides additional information about a node as soon as the mouse hovers over a node either in the session view or in the task view. A thumbnail of the web page is shown, as well as its title, URL, and the number of visits to the related page during this SG-session. It has been shown that thumbnails improve the user's ability to quickly re-identify an already visited page (Czerwinski et al. 1999; Woodruff, Faulring, Rosenholtz, Morrision & Pirolli 2001). The detail view panel is also used to display search results after querying the SessionGraphs history.

This GUI was created in close dependence on a valuable guideline for designing visual GUIs, Ben Shneiderman's *Visual Information-Seeking Mantra*: "overview first, zoom and filter, then details-on-demand" (1996). Accordingly, the SessionGraphs GUI is designed to provide an *overview* first, on each level of access: the task chooser provides an overview of all SG-tasks and SG-sessions, the task view of the current SG-task's SG-sessions, and the session view of the current SG-session's nodes. Secondly, the GUI enables the user to *zoom in* on items of interest – by select-

¹ More on the aspects of tight coupling can be found in (Ahlberg & Shneiderman 1994).

ing an SG-task or SG-session in the task chooser, the user zooms in on a specific SG-task or SG-session. Thirdly, it allows information to be filtered – to view only information related to one specific SG-task – and to present *details-on-demand*. Hovering over nodes activates the detail view while clicking shows full detail in the connected web browser – the page’s actual content. Additionally, Shneiderman recommends providing a *history of interface actions* that can be undone again. SessionGraphs contains a related undo / redo function – all modifications the user applies to the GUI (such as coloring, deleting, or dragging a node) are undoable / redoable again.

As previously mentioned, the approach was designed before Bederson’s mapping of Csikszentmihalyi’s notion of *flow* (1991) onto GUIs. Still, when the approach is viewed under this perspective, it can be ascertained that it satisfies the major characteristics demanded by Bederson (2004):

(1) It *challenges and requires skill* in the sense “that it is possible to learn a tool well enough so that it becomes almost an extension of one’s body, like a hammer” (ibid.) by enabling the acquisition of skills on all three levels identified by Anderson (1995). On the first, or *cognitive level* of skill acquisition, the SessionGraphs tool is easy to learn by following the instructions offered, for example, using an online tutorial or instructional videos. The second, or *associative level* is supported by visual feedback from the interface itself, for example, by the animated, interactive behavior of the graphs and menus. The third or *autonomous level*, characterized by *automated, rapid skills* and minimal cognitive involvement, is supported by keyboard shortcuts for most functions that allow commands to be executed independently from the interface feedback.

(2) Furthermore, the approach allows a person to *concentrate better and avoid interruption* by providing non-modal dialogs and showing the GUI to the side of the browser, enabling the user to ignore it whenever desired in a similar manner to the Windows XP “bubbles” linked to the system tray.

(3) It enables the user to *maintain control* by being non-adaptive albeit highly adaptable. Figure 3-11 shows the “settings” dialog that allows many settings to be

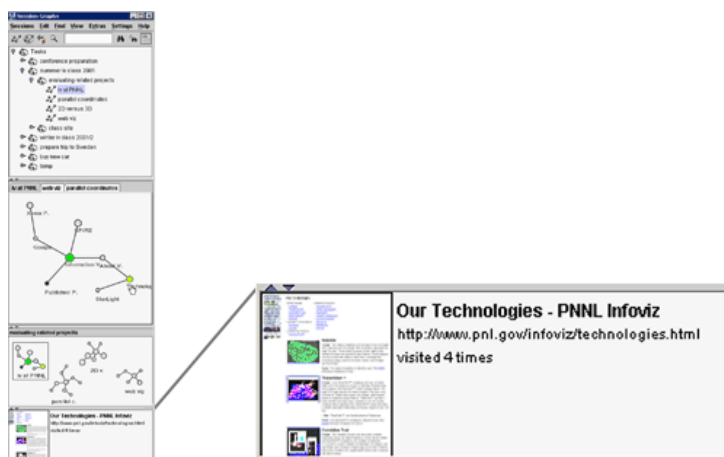


Figure 3-10: SessionGraphs GUI (5) - the detail view.

configured based on individual preferences. Most color values (views' background, node body and border, edges) can be adjusted to a preferred color combination and the default node sizes, as well as further behavioral aspects can be defined. The user can create and then switch between different self-defined and named sets of settings. The setting that was used on system shutdown will be applied upon the next launch.

(4) It supports *speed and feedback* by being designed for high responsiveness, giving the user immediate feedback on almost any interaction including page visits executed with the web browser. Bederson (Czerwinski, Horvitz & Cutrell 2001) points out that in order to maintain *flow* in Csikszentmihalyi's sense, it is important to "allow the person to clearly set goals and receive feedback about their progress towards those goals". This is accomplished, first, by the undo / redo support, and second, by the possibility for explicit task representation and the visualization of sessions, which give the user immediate clarity about the parts that are already accomplished. For the case where the user decides to create a partial task structure before executing them, clarity is also available for the parts that still have to be done.

(5) Finally, the *transformation of time* – or subjectively perceived duration of a task – seems to be a reliable indicator of flow. Czerwinski et al. (2001) showed that web-based activities, when interrupted and then later resumed, tend to be overestimated in duration, whereas tasks that are accomplished successfully all at once tend to be underestimated. Since the overestimation in time could be based on design flaws that interrupt the user's flow, they conclude that "Tasks for which participants overestimated the length should be considered high priority tasks to make more usable by the browser design team" (Czerwinski et al. 2001: 169). Certainly, the additional effort of task and session maintenance, as well as the perception of the graphs can be considered interruptions of the actual activity. However, the assumption is that, for important and more complex activities, this is a worthwhile investment facilitating the further proceeding.

Before presenting the prototypical implementation, a scenario will be used to outline an exemplary use of the tool. It will be described how the identified seven situations of revisitation are supported by SessionGraphs.



Figure 3-11: The SessionGraphs configuration settings dialog. Allows the user to define (several sets of) individual preferences.

3.2.4 Scenario of Use - Addressing the Seven Situations

This scenario illustrates how the SessionGraphs concepts address the seven critical situations of revisitation (chapter 2.3.3), extending the example of Hera, the professor who is planning her information visualization class and a trip to Sweden (compare page 22). It compares how Hera could go about accomplishing the upcoming tasks either with a current browser (A), or with the SessionGraphs tool (B).

A) Hera with a Common Browser

To prepare for the upcoming class, Hera is looking for exciting projects on websites that publish research about information visualization. As she explores the Pacific Northwest National Laboratory's website for new visualization projects, she investigates several potentially interesting subbranches in different windows. After gaining an overview, she wants to go back to two of the subbranches to explore them in more detail. She attempts to use the back button, but realizes that this does not bring her back far enough, since the related window was newly opened amidst the path and the back button does not support her going back to another window. She has difficulties finding the two pages in the histories of the open windows (type 1: Back to Another Branch or Window Revisit).

She spends about an hour exploring different sites and topics in order to find projects that might be interesting for her class. Her next task is to create a reading recommendations list. For this purpose she has to go back to the respective publication pages. However, this is not as easy as she wished, since she only bookmarked a few of the pages. Other pages she simply cannot find again using the URL entry's autocompletion feature, since too many URLs with the same beginning are listed. Consequently, she finds herself using a search engine, navigating to home pages and following links to publication pages exactly as she did before. This unnecessarily results in almost the same effort of navigation as the previous hour (situation 2: Parallel Session Revisits).

Hera also uses the web for other tasks simultaneously, for instance, looking for a hotel for her Sweden trip. She visits several hotel websites to compare rooms and prices. In the middle of her work she is called to join a meeting. She finally finds time again to resume her search three days later. Unfortunately, all the information she painstakingly gathered earlier is lost – she starts over by revisiting the hotels' home pages again, locating all the detail pages about rooms and prices from scratch (situation 3: Session-Resumption Revisits).

The semester starts and Hera wants to review a specific detail about one of the research projects on the evening before she gives her lecture. She goes to the institution's home page, locates the project again, but cannot find the page which

mentioned the desired detail in a reasonable amount of time (situation 4: Rediscover Revisits).

Every week throughout the semester, Hera looks at her students' pages that document the advancement in their own projects. To do so, she first uses a toolbar bookmark that brings her to the class website and then uses link following, up to four levels deep, to locate a distinct student's page. Cumbersomely, she has to move up and down her site's hierarchy visiting all the students' pages, always using the same or similar paths (situation 5: Same-Path Revisits).

Half a year later, Hera is asked to teach the same class again. In preparing, she revisits most of the research project sites she originally found in order to find out about the projects' current status and to search for new projects. Unfortunately, she has to redo the entire search again, since she had only made a few bookmarks. This entails a search from scratch, as all the browser's history tools (such as URL-autocompletion, breadcrumbs, or history list) in the meantime no longer offer the data (situation 6: Similar-Task Revisits).

Finally, while preparing for the new class she has a teaching assistant join her. She wants to inform him about several interesting research projects that he should become familiar with. To do so, she copies single URLs from the browser's URL field to her E-Mail client and sends him an email. This takes longer than she thought, so she is not able to include all the interesting pages since her time is limited (situation 7: Group-Exchange Revisits).

All in all many unnecessarily complicated, potentially frustrating procedures.

B) Hera using SessionGraphs

By using SessionGraphs, Hera could handle her tasks in a more organized and efficient manner. Let's assume she immediately starts exploring the Pacific Northwest National Laboratory's site. All related visits are represented as nodes in a corresponding session visualization (Figure 3-7). She can use any node to immediately backtrack and explore further branches on this site even if located in different windows (situation 1: Back to Another Branch or Window Revisit).

Hera arranges the related SG-tasks whenever she likes. She creates a new SG-task "summer IV class" and a sub-task "evaluating related projects" (see Figure 3-6). All the explorations of related sites and topics are added to this sub-task. For example, she created SG-sessions about "parallel coordinates" and about "2D versus 3D" approaches.

To create the reading recommendations list, she has to go back to the publication pages. To do so, she simply uses the task view (Figure 3-9) which shows miniature visualization of all related sessions. Since Hera is familiar with SessionGraphs, she had colored the nodes that represent publication pages in orange immediately after accessing them. Now she simply revisits all the orange pages (situation 2: Parallel Session Revisits).

When Hera gathers information about Swedish hotels, she is creating a new SG-session. Consequently, interruptions are not a problem. Three days later, the complete path is still available and she can resume the SG-session exactly at the point where she stopped (situation 3: Session-Resumption Revisits).

To rediscover detail information before her lecture, Hera simply reopens the session visualization of the earlier SG-session. Instead of circuitously exploring the complex site again, she hovers over several nodes in the visualization until she finds the desired target (situation 4: Rediscover Revisits).

During her weekly visits to the students' pages, the look-ahead graphs become active automatically (see Figure 3-12), thus giving immediate access to any earlier SG-session in which she traversed by the same path. Instead of opening several unnecessary navigation pages week after week, she simply selects the desired target nodes in the look-ahead graph (situation 5: Same-Path Revisits).

When Hera is asked to teach the same class again, she simply selects the earlier SG-task "evaluating related projects", copies it, and uses the copy to revisit all the interesting project sites (situation 6: Similar-Task Revisits).

To pass on information to her teaching assistant about interesting research pages, she simply selects the SG-sessions with the related information and selects "mail a session to a friend" from the menu. Her assistant, who also uses SessionGraphs, can then revisit all related pages.

These examples illustrate how the SessionGraphs approach provides benefits in specific situations.

3.2.4.1 How the 7 Situations of Revisitation are Supported

The following overview gives a more complete view of the applied concepts used to support the seven situations of revisitation (see chapter 2.3.3).

Situation 1: Back to Another Branch or Window - Revisit

The aforementioned critical situations of short-term revisitation are supported in the SessionGraphs approach by the concept of session-based visualization. A session visualization gives the user direct access to all earlier pages, including hub and spoke, and navigation and search pages – no matter in which window or tab they were visited. When the back button fails either because the desired target page has been popped from the stack or it was visited in another window, the session visualization will still hold a reference to it. Ariadne (compare page 1) could have easily revisited the desired page on mazes – it would have been represented at the end of a spoke of the current hub.

Situation 2: Parallel Session Revisits

Using separate SG-session visualizations for each concurrent session, the SessionGraphs user can easily distinguish between different parallel activities. He or she can

immediately switch back to any other parallel session either by interacting with the session view tabs (to navigate within the last three sessions), the task view, or the task chooser. Once the desired session visualization is visible, any represented page can be opened in the web browser. Compared to current browsers that provide multiple windows and tabs as a workaround, the user can label sessions in a meaningful way, see the history of a session immediately and not just the last visited page, use the task hierarchy to separate non-related tasks, and still use multiple windows and tabs. To re-find the Retsina wine offer, Bacchus (compare page 2) could have simply switched from the map search session and opened the last session tab visualizing all the searches on wine, even if they are spread over different browser tabs.

Situation 3: Session-Resumption Revisits

In contrast to current browsers, the SessionGraphs tool stores complete history information for days, weeks, and months. The user can reconstruct all earlier sessions in full detail. The concept includes the ability to reconstruct (rearrange) all the browser windows and tabs to its previous state, including window histories and positions.¹ Once an SG-session is visualized, the user can immediately re-access all related nodes, the complete earlier path, and resume the SG-session. Chloris, who was called to her supervisor, could easily have resumed the literature search session four days later. All pages previously visited would be visualized in the related session visualization just as she had left it (compare page 2).

Situation 4: Rediscover Revisits

The SessionGraphs history approach has two primary advantages compared to current browsers regarding the revisitation of single, hard-to-rediscover pages. First, it saves all visited pages without deleting them. Thus, the user can be sure that an item can be found in the history. Second, since the user maintains the task hierarchy, the related session can easily be found by navigating through the task hierarchy. It should be more likely that a user remembers the task that comprised the last visit to a desired page than the exact date, frequency, or – in many cases – domain name. Still, if navigating the task hierarchy is not successful, the hierarchy itself can still be searched exactly like a browser's history list (in the implemented prototype only textual search was executed, no filtering or grouping). If Daedalus had used SessionGraphs to find the image again (compare page 2) he could have simply navigated through his task hierarchy down to the task where he initially discovered the picture. Using the thumbnails attribute, he would quickly locate the correct node within the related session visualization.

¹ This was only partially achieved in the implemented prototype, since it would have required a closer connection to the browser software.

Situation 5: Same-Path Revisits

The SessionGraphs system detects when a user reuses familiar paths and offers him earlier session visualizations (in the form of the so-called *look-ahead graphs*) to directly revisit a potentially distant target page, and therefore reduce unnecessary, time-consuming intermediary navigation pages. To do so, the system analyzes the user's path. Whenever it detects a sequence of three visits that also appeared in one or more earlier SG-sessions, it offers him all matching SG-sessions for potential reuse. These *look-ahead graphs* (see Figure 3-12) – miniaturized session graphs from earlier SG-sessions – temporarily replace the task chooser. The user can use them in five different ways – either by directly clicking a node in one of the look-ahead graphs or by using one of the four buttons on the right (merge, copy, switch, direct). These buttons can be used to merge a copy of the earlier SG-session into the current SG-session, to merge both SG-session originals, to switch to a copy of the earlier SG-session, or to switch to the earlier SG-session directly, not to a copy of it. Switching to a copy has advantages when the original should be kept untouched for other pos-

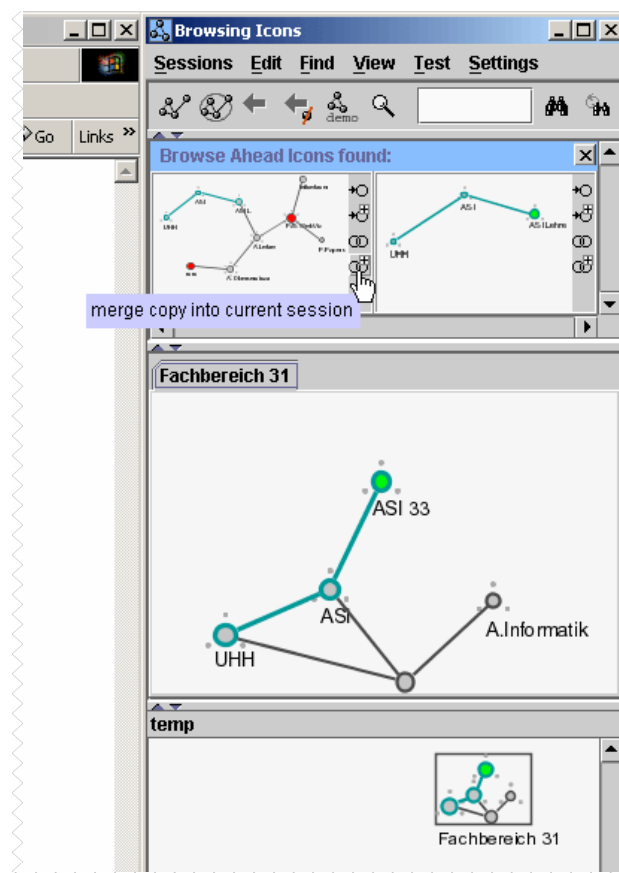


Figure 3-12: Look-Ahead-Graphs to support same path revisits.

The task chooser is temporarily replaced by small versions of session graphs that contain the same sub-path as the current SG-session (marked with blue highlight). They are automatically offered and provide shortcuts to potential target pages.

sible occasions. After any interaction, the look-ahead-graphs will disappear and be replaced again by the task chooser. These possibilities offer adequate choices for many different situations of long-term revisitation. For example, whenever Echo would start to repeat an earlier path (compare page 2), the look-ahead graph(s) would pop up giving her direct access to the desired page or at least the desired session.

Situation 6: Similar-Task Revisits

The SessionGraphs' task concept inherently contains the grouping according to the user's tasks. Thus, whenever the user needs to do something similar again, requiring several of the earlier sources, he or she can continue simply by copying all related earlier SG-tasks or SG-sessions (by menu, shortcut, or drag and drop) including all earlier path information. The end result is a deep copy of an SG-session or SG-task that allows one to reuse all the references independently, rename nodes, SG-sessions and SG-tasks, and remove and visually reorganize specific nodes, however needed, without modifying the original task and session information. Using SessionGraphs, Phaeton could have made a copy of his earlier task from six months before (compare page 3) and then started immediately to reuse some of the earlier session information according to his present needs.

Situation 7: Group-Exchange Revisits

The SessionGraphs prototype provides a very basic means for exchanging history information among group members. Session information can be saved and emailed so that a receiver also using the tool is able to view, use and modify the received representation. This comes close to the vision of Vannevar Bush who pointed out the importance of being able to exchange information about trails between different users (Bush 1945: 45). Using this approach, Gaea could have easily mailed the interesting history information to her two coworkers (compare page 3).

This overview presented the concepts and ideas underlying the SessionGraphs approach to directly address all seven situations that were identified as critical for well-supported revisitation. The next section will describe how these concepts were technically realized in prototypical software.

3.3 Prototypical Realization

To study specific aspects of the SessionGraphs concept, several prototypes of different complexity and purpose have been built following the typical phases of a software engineering project (see Figure 3-13). Each phase went through various iterations. Voluntary users and usability experts were actively involved in the process at different stages. They gave valuable feedback, which influenced many decisions on the way to the final version of the prototype.

The SessionGraphs concept was iteratively derived starting with a number of hand-drawn and computer-aided sketches (see Figure 3-14) that were discussed with colleagues and students who also had a background in web navigation and web usability. Thereafter, several small throwaway prototypes (see Pomberger & Blaschek 1996: 3-5), comprising different GUI approaches and alternative visual behaviors, were used in a process of *explorative prototyping* (see Floyd 1984) to elaborate and specify the initial idea. They resulted in a concise specification of functionalities and GUI design. One important result from discussing the approach with colleagues was to provide the complete SessionGraphs GUI within one clearly structured window rather than in different independently moveable windows per SG-session as initially intended (see Figure 3-14).

The design phase consisted of the definition of entities and relations that model the application domain. Different technical solutions were considered during the early implementation phase for specific aspects (experimental prototyping, see Floyd 1984). The final prototype – “prototype A” – was built once the most important technical decisions were solidified. It was then used for evaluation in the longitudinal SessionGraphs study and is referred to in this text. As a complete prototype (Pomberger & Blaschek 1996: 5), it comprised the most important functionalities of the SessionGraphs concept. A predecessor of prototype A was used to conduct the controlled experiments described in chapter 4.2. Between the controlled experiments and the longitudinal study, different modifications were introduced based on the feedback received during these experiments and during subsequent informal feedback loops with experts and students. This process of *evolutionary prototyping* finally led to prototype A.

One of the modifications was to support toggling the visualization between logging and not-logging the path. Several users mentioned in the controlled experiments and informal pre-studies their desire to be able to temporarily switch off logging in order to constrain the recorded visits and sessions to subjectively meaningful ones. This comes close to Rosenberg’s (1996: 25) demand: “At its simplest, such a gathering interface would have commands ‘begin episode’ and ‘end episode’ similar

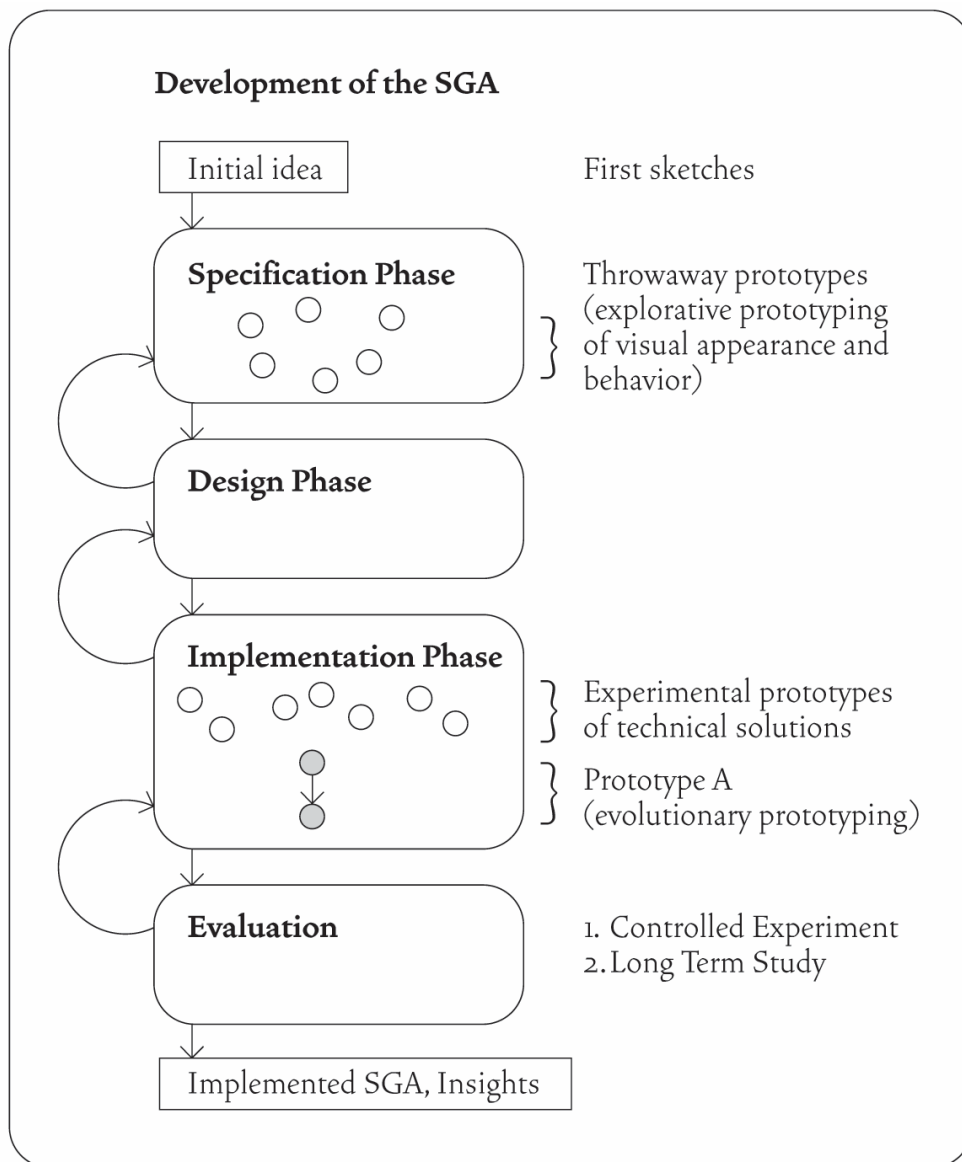


Figure 3-13: SessionGraphs development - phases and prototypes.
 "SGA": SessionGraphs Application.

to the 'record' and 'stop' commands common in numerous application program macro recorders".

Thus, three different modes of action were introduced to prototype A before carrying out the longitudinal study: `SESSIONGRAPHS-ACTIVE` (tracking and visualizing the path as usual), `SESSIONGRAPHS-SUSPENDED` (visible SessionGraphs GUI, but suspended tracking and visualizing), and `PLAIN-INTERNET-EXPLORER` (hidden SessionGraphs GUI, neither tracking nor visualizing, but still logging the visits for study purposes). This third mode was provided for users who wanted to temporarily use the web with a plain web browser, without any SessionGraphs enhancement.

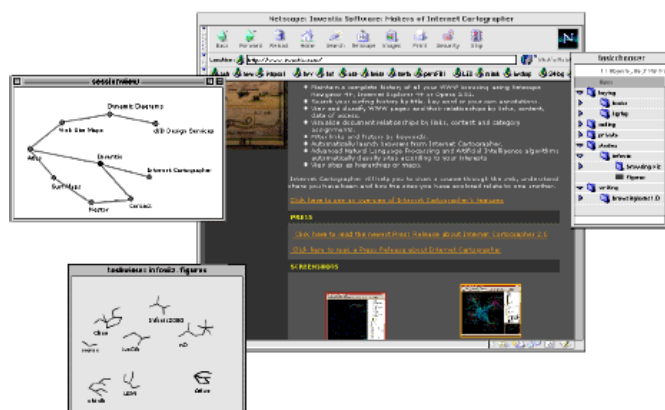


Figure 3-14: Early SessionGraphs prototype. Interviews revealed that people prefer a single history window that contains the three views here spread over three different windows.

3.3.1 Software Architecture

The following requirements for the communication between web browser and prototype guided the technical decision process:

Browser to prototype:

- Notify prototype immediately about every visit (URL),
- provide source code of the current page for further data extraction,
- Further, “nice-to-have” requirements:
 - tell prototype the window ID of every visit,
 - tell the prototype how the user interacted with the browser, e.g. the used history mechanism,
 - let the prototype know, which browser window currently is on top.

Prototype to browser:

- Load a certain URL (into a specific or new browser window),
- bring a certain browser window to the front.

Microsoft Internet Explorer seemed to be a suitable choice since several technologies existed for extension, communication and modification. Internally, Internet Explorer is composed of several components (MSDN 2007) the interaction of which is governed by the Component Object Model (COM), which also enables component reuse and extensibility (ibid.). It allows the browser’s *appearance* to be extended, for example, by customized tool bars and explorer bars. It also enables the types of displayed *content to be* extended, e.g. by ActiveX Controls and Active Docu-

ments, and it allows *hosting and reusing entire components* in external applications, e.g. by using the components *shdocvw.dll* (the WebBrowser control that provides history and navigation functionalities) or *mshtml.dll* (the parsing and rendering component often referred to by its code name, “Trident”). Possible technical solutions have been evaluated in a phase of intense testing and in close exchange with other colleagues¹.

The choice of Java seemed to be a reasonable decision, based on Java’s high degree of object orientation, the possibility of straight-forward prototyping, easy implementation, and relative platform independency. Throughout the project, Java versions 1.2 to 1.4.1 were used for implementation.

The dynamic visualizations of SessionGraphs required proper support of graphics. A toolkit capable of supporting such visualizations on a high level is the Jazz toolkit for zoomable Java2D Graphics (Bederson et al. 2000; Bederson & McAlister 1999; Bederson 2002). This toolkit was chosen, since it provided ready-to-use graphical classes that modeled graphics on a much higher level than the standard Java 2D package. My visit at HCIL allowed close cooperation with the developers, which was highly adjuvant for the prototypical implementation.

Since the Jazz toolkit required a Java 2 environment, and the Microsoft Java Virtual Machine (MSJVM) used by Internet Explorer only supported Java 1.1, the choice was made to implement the prototype as a stand-alone Java application next to Internet Explorer, using a separately installed Java 2 virtual machine. Java applets could not be used, since they run integrated within the browser using the MSJVM. Besides the advantages of being able to use Jazz, this solution made it easy to provide one visualization for all browser windows, which would not have been possible using separate applets. Moreover, the loose coupling between the browser and prototype would make the use of future browser versions easier compared to a modified browser version. This decision, however, meant that technical solutions had to be found to support communication between the browser and prototype.

The final “prototype A” was entirely built as a Java 2 application, acting as a proxy between the web and Microsoft Internet Explorer. It uses the information about the web pages requested by the browser (HTTP requests) in order to trace the user’s path.

Figure 3-15 shows the technical setup of prototype A and the environment of third-party-software. The prototype communicates with the web browser (on the

¹ Helpful support was given by GroupLab’s developer Shaun Kaasten, and Scone developers Harald Weinreich, Hartmut Obendorf, and Torsten Hass. Thanks!

left) via the Scone framework as HTTP proxy (lower middle).¹ The Scone framework establishes the connection to the web (lower right). The prototype itself is implemented as a Scone Plugin that registers to Scone to be informed about any visited web page.

Other third party components accomplish visualization and interaction features. The internal module **BrowserControl** enables the prototype to load a specific web page directly into the browser. A second way to accomplish this would be to use an invisible applet (“Scone Applet”) that is automatically integrated into each page’s source code by Scone. This second method allows for a more exact control of the browser, but is not always available as explained in more detail below. The right side shows of Figure 3-15 five components that make use of Java I/O to guarantee persistency for the history information, study logs, user preferences, etc. This technical setup promised to meet the requirements stated above. The single participating components are introduced in the following.

3.3.1.1 Integrated Frameworks and Libraries

Scone Proxy

The Scone framework (Weinreich 2006; Weinreich, Buchmann & Lamersdorf 2003; Obendorf, Weinreich & Haß 2004) for development and evaluation of web enhancements offers several promising features that were used in this project. First, it provides a complete proxy and handles all the communication between the web and a web browser. Second, it inserts a layer that models and encapsulates requested objects and browser events. These representations can be immediately used by any **Scone Plugin**. The SessionGraphs prototype is implemented as such a plugin. Harald Weinreich who is the head of the Scone project, was a close colleague of mine and the cooperation meant both that rising questions could be quickly answered by the Scone team, but also that SessionGraphs was a welcome application for them that may reveal shortcomings of the current Scone implementation. For the period of the cooperation, Scone went through its own versions 0.7 to 1.1.

Scone (Weinreich et al. 2003) is a Java-based framework published under the GNU General Public License and built upon IBM’s Web Based Intermediaries concept (IBM 2007). Scone plugins are written in Java; they are notified about any http

¹ HTTP proxies act as intermediaries between client and server and can modify, control and adapt the transferred content on the fly. Requests are received from clients and forwarded to servers - responses are passed back in the same way. The Scone proxy is used in order to inform the SessionGraphs application about the requested items. A proxy can also be used as a transcoder to modify content, as a child filter, as a document access controller in large corporate settings, as a security firewall, as a web cache to store local copies and to reduce traffic, as a surrogate to accelerate a server, as a content router, or as an anonymizer to secure privacy issues (Gourley & Totty 2002: 131ff).

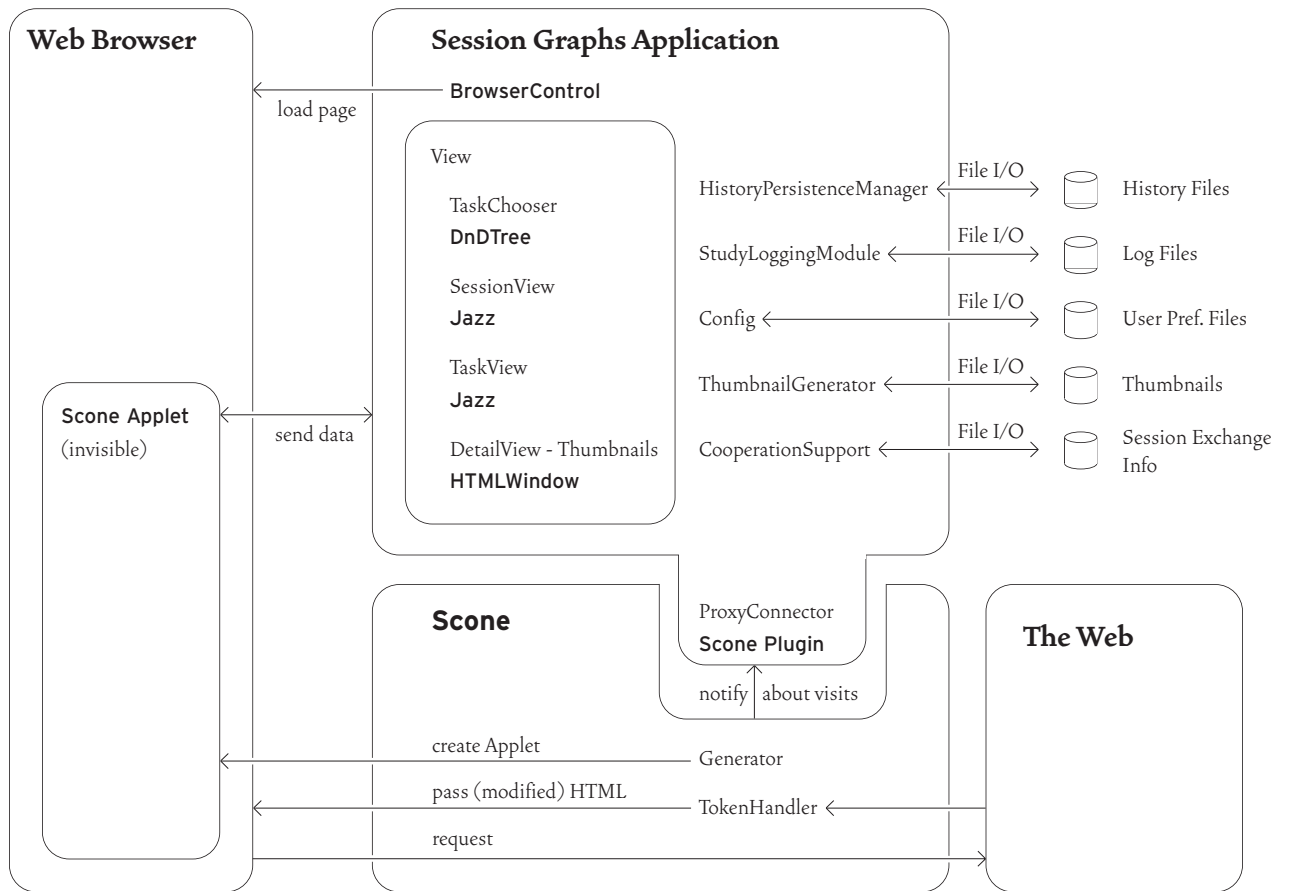


Figure 3-15: Environment of prototype and third party software.
 The figure shows how the SessionGraphs application is integrated in the context of Scone, the web, and a web client. Sans serif font indicates third party products.

requests that the browser submits to the web. Scone allows creating **Generators** and **TokenHandlers** that can modify and enrich the source code of HTML pages by additional code or applets before they are displayed. This ability was interesting e.g. for sending load commands to specific browser windows via an invisible applet integrated into an HTML source (Scone Applet, Figure 3-15). HTML pages are represented as Java objects (e.g. **NetNode**, **HtmlNode**) and detailed information is provided about browser events (e.g. “link followed”, “back button used” or “new window opened by user”) derived by certain heuristics. The access tracking module allowed tracking the user’s interactions with the browser and the web pages.

As a Scone plugin, the prototype extends the class **scone.Plugin** and acts as an observer to **scone.netobjects.AccessCache**. It is notified about new visits by a **scone.netobjects.AccessEvent**. This event contains the URI and title of the accessed web page, related timestamps, a window identifier, the load method, the number of internal and external links and other information about the web page. The prototype internally processes these values to create the desired visualization.

Jazz Graphics

In contrast to most 2D graphics toolkits, the Jazz toolkit for zoomable Java2D Graphics (Bederson et al. 2000; Bederson & McAlister 1999; Bederson 2002) provides not only a renderer, but also a hierarchical scenegraph that organizes all visible objects, cameras and transformations, as known from 3D systems. This enables the programmer to model a view on a higher level than that offered by the Java2D API. It provides cameras, layers, basic visual elements, affine transformations, semantic zooming, selection and modification handlers, as well as multiple visual representations of the same object. These abilities provided an ideal basis for the designed visualizations.

Jazz distinguishes between two kinds of scenegraph objects, **ZNodes** and **ZVisualComponents**. **ZNodes** (groups, transform groups, leaf nodes etc.) are located at a specific position within the scenegraph hierarchy and store inheritable information, such as transformations. **ZVisualComponents** only determine the visual appearance of an object independent of transformation or hierarchical position. Figure 3-16 shows a simple Jazz scenegraph in which two **ZPolylines** and one **ZRectangle** - both **ZVisualComponents** (rectangular outline) - are organized within a scenegraph hierarchy using **ZVisualLeafs** and **ZGroups**, which are both **ZNodes** (oval outline). A **ZCamera** looks onto a **ZLayerGroup**, which groups the entire visible scene. The architecture allows strictly separating appearance from structure and manipulation. Jazz supports an event mechanism based on event listeners similar to Java. Listeners can be attached to any node in the hierarchy and receive the dispatched events of the entire subtree. Based on the application's needs, a listener can be used to handle events very locally or rather globally. The application of Jazz in relation to the different SessionGraphs views is described in the respective chapters.

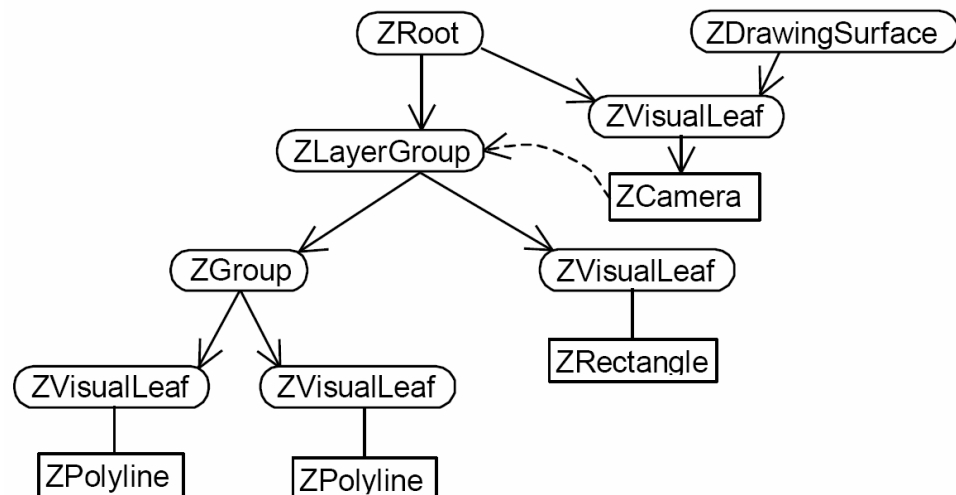


Figure 3-16: Structural diagram of an exemplary Jazz scenegraph. ZVisualComponents are shown using a rectangular outline, ZNodes with an oval outline.

Drag and Drop Support

Drag and drop (DnD) interaction was necessary for the simple and quick manual reorganization of SG-sessions and SG-tasks in the **TaskChooser**. Since the DnD API introduced by Java 2 (java.awt.dnd in combination with java.awt.datatransfer, see De Lisa 1999a) creates unnecessary high complexity simply by the sheer number of classes and interfaces and several circuitous message-passing methods, the freely available **DnDTree** classes developed by Rockhopper Technologies were used instead (De Lisa 1999b). They are still based on the Java DnD API, but enable the programmer to work around the given deficiencies. They encapsulate a DnD supporting version of the **JTree** that is used for the **TaskChooser** implementation.

Thumbnail Generation

Since Java does not allow thumbnails of web pages to be created, another third party product was required. The HTMLWindow API (Heistermann 2007) wraps the autonomous web browser “HorstScape”. This browser is used by **sga.util.ThumbnailGenerator** to create an image object of any specified web page. HorstScape produces commonly known miniatures of a web page’s visual appearance. It was outside the focus of this thesis to further improve the thumbnails by techniques like dogears (compare Cockburn et al. 2003) or by enriching them with textual phrases as suggested by (Woodruff et al. 2001).

Figure 3-17 shows how Heistermann’s API is integrated into the prototype’s code; **HTMLWindow** extends **JScrollPane**, **HTMLPane** extends **JPanel**. In line 8, the referenced HTML page is rendered using a predefined virtual browser window size; in line 9 the Java **Image** is created. Lines 10-12 are responsible for scaling the image to the desired size. Thumbnail creation can take up to several seconds, since the complete HTML source and embedded media are downloaded for rendering. In order to not interrupt the running program, thumbnail creation is delegated to separate threads that are set to minimal priority encapsulated by **sga.util.ThumbnailGeneratorThread**.

To accelerate the display of thumbnails and reduce processor load and net-

```

1 import horst.HTMLPane;
2 import horst.HTMLWindow;
3 public static Image generateThumbnail(URL url) {
4     Image thumbnail;
5     HTMLWindow hw = new HTMLWindow();
6     HTMLPane hp = hw.getHTMLPane();
7     Frame f = new Frame(); f.addNotify(); f.add(hw);
8     hp.syncOpenPage(url);
9     Image img = hp.getPageImage(1, 800, 1000);
10    thumbnail = img.getScaledInstance
11        (-1, ConfigMain.getCurrentConfig().THUMB_Y,
12        Image.SCALE_SMOOTH);
13    return thumbnail;
14 }

```

Figure 3-17: Thumbnail generating code.

The class **sga.util.ThumbnailGenerator** uses the HTMLWindow-API by Heistermann.

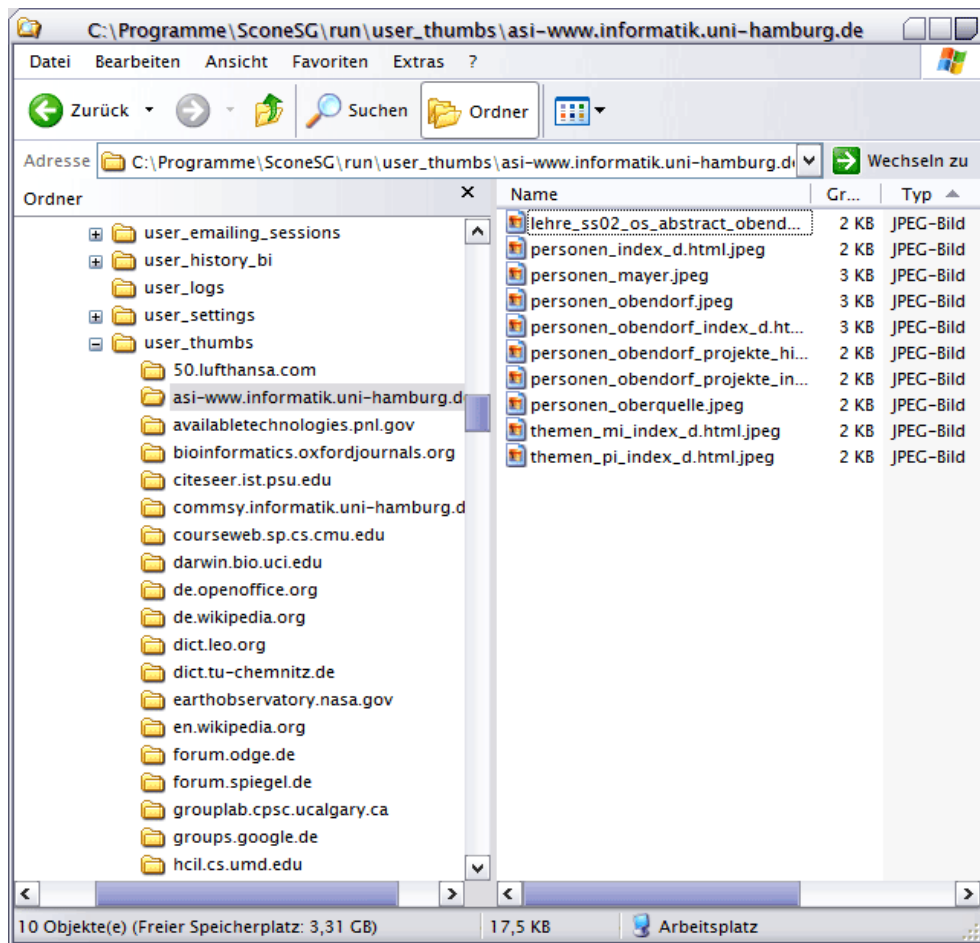


Figure 3-18: Directory and file structure for local thumbnail storage.

work traffic, every thumbnail is stored locally once it is created. Whenever available, a local version is used. A generator thread is only instantiated when no local thumbnail exists. The thumbnails are JPEG image files, usually between 1 and 4 KB in size. To effectively organize them in local directories, the filename and path is created on the basis of their URL. The directory name is automatically derived from the host part, the filename from the path part – see Figure 3-18. A thumbnail of page www.bwz.de/web/news/main.htm, for example, is stored in the file “user_thumbs/www.bwz.de/web_news_main.htm.jpeg”. This guarantees that any URL can easily be found again quickly even if a large number of local thumbnails exist. This method could be further improved by reflecting changes in dynamically updated web pages and querying parameters also contained within the thumbnails.

Controlling the Browser

While being informed about visited pages was one important task – for which the proxy was a good solution – controlling the browser, i.e. forcing the browser to show certain pages, was the second important task.

Ideally, the browser would be completely controllable. Not only would it be possible to force it to display a certain page, but also to load the page in a specific window or tab and to place the window at a specific position. This was not possible based on the chosen proxy solution. The final prototype offers two alternatives for controlling a web browser.

The first method offers more control and is based on Java applets. It allows both the page that should be displayed and the window in which it should be displayed to be controlled. The prototype uses this method whenever a web page is revisited within an SG-session and the window in which it was visited before is still open. Scone's **acesstracking** package is then used to load the page directly into the browser window in which it was displayed last. For this purpose, Scone enriches every displayed HTML page by an invisible Java applet using **AppletConnector** and related helper classes. A socket connection between this applet and the Scone plugin enables the exchange of information and commands in both directions. The **SessionGraphs** prototype uses the **sendToApplet()** method to display URIs in the addressed web browser window. For example, **appletconnector.sendToApplet("openURL", url, "_self")** loads the web page specified by *url* into the same browser window that the applet is located in.

The second method offers less control and is based on a modification of the **BrowserControl** class written by Steven Spencer (1999). It controls only the page that should be displayed, but not the window in which it should be displayed. Its advantage is that it works very reliably and does not require any browser window to be open before. A disadvantage is that it restricts the possible platforms to Unix and Windows. The command **sga.utils.BrowserControl.displayURL(url)** loads the selected page in the frontmost browser window without leaving a possibility to further specify the window – or a new window if no other window is open. The prototype uses this second way whenever no **AppletConnector** is available – i.e. usually whenever the related window was closed before or it is the first visit to this URI in this SG-session.

3.3.1.2 Internal Architecture

The software-technical design and implementation of the prototype was influenced both by the need to develop a manageable and modifiable prototype in a one-person project, and by the desire to implement the most important aspects of the conceptual approach. One requirement for the prototype's architecture was that it should allow different alternative technical and conceptual approaches to be easily explored, e.g. different layout and animation algorithms, different heuristics for determination of SG-session borders or different behaviors of the graphical interface. Thus, object orientation seemed to be an appropriate strategy both for design and implementation. The chosen degree of modularization should allow local modification and ex-

tension without influencing the remaining code. The overall complexity was kept reasonably small and module cohesion and module coupling were balanced to the extent possible (see Pomberger & Blaschek 1996: 53ff). The advantages of object orientation – extensibility and reusability (Pomberger & Blaschek 1996, page 6) – were very welcome. Extensibility was important for refining the prototype; reusability was helpful since specific parts –for example mouse listeners and event handlers – could easily be reused within the application itself.

The software architecture of the SessionGraphs prototype follows two common paradigms – the *model-view-controller pattern*, MVC, (Reenskaug 2003; Johnson 2004; Gamma, Helm, Johnson & Vlissides 1996: section 1.2) and the *logical 3-tier architecture* (Marston 2004b; Eckerson 1995; Counihan, DeVoge & Bass 2005) – see Figure 3-19. Applying these paradigms during the design and implementation phases helped to clarify fundamental relationships between the single modules. The MVC pattern distinguishes between parts that take care of internal processing (the MODEL), UI output (the VIEW), and user input (the CONTROLLER). The *logical 3-tier architecture* separates the *data layer*, *function layer*, and *presentation layer*. Both approaches have been combined systematically in the SessionGraphs prototype.¹ In the resulting internal architecture (see Figure 3-19), the MODEL of the MVC contains both the *data* and the *function layer* of the *3-tier approach*. The *presentation layer* of the *3-tier approach* contains both the VIEW and the CONTROLLER of the MVC pattern. The architecture was developed in a top-down design phase and further refined during the early stages of experimental and evolutionary prototyping.

The *data layer* stores the internal representations of SG-sessions and SG-tasks, and a hierarchical structure that relates both. A **SessionRepresentation** keeps a visits list of all visits that occurred during an SG-session (modeling VL, page 81), as well as a reference to the related session graph (SG, not depicted in the figure). The hierarchical organization of SG-sessions and SG-tasks is accomplished by the **TaskTreeModel** using **TreeNode** containers (**TaskNode**, **SessionNode**), which are utilized to create a hierarchical structure that can be presented to the user. They reference **Session-** and **TaskRepresentations** that hold the meaningful content.

The *function layer* comprises both external functionality, accessible via the GUI, and application-internal functionality. The **BrowserConnector** (including **BrowserControl** and **Scone Applet**) enables the SessionGraphs prototype to load a certain page into the browser once requested by the user. The **ProxyConnector** informs the prototype about new pages loaded by the browser; the **SessionController** organizes the determination of SG-session borders using a certain heuristic module; the **HistoryPersistenceManager** archives history and visualization data; **ConfigMain** manages user definable persistent settings; the **StudyLoggingModule** is responsible for tracking user activities in the studies.

¹ See (Marston 2004a) for the combination of MVC and 3-tier architecture.

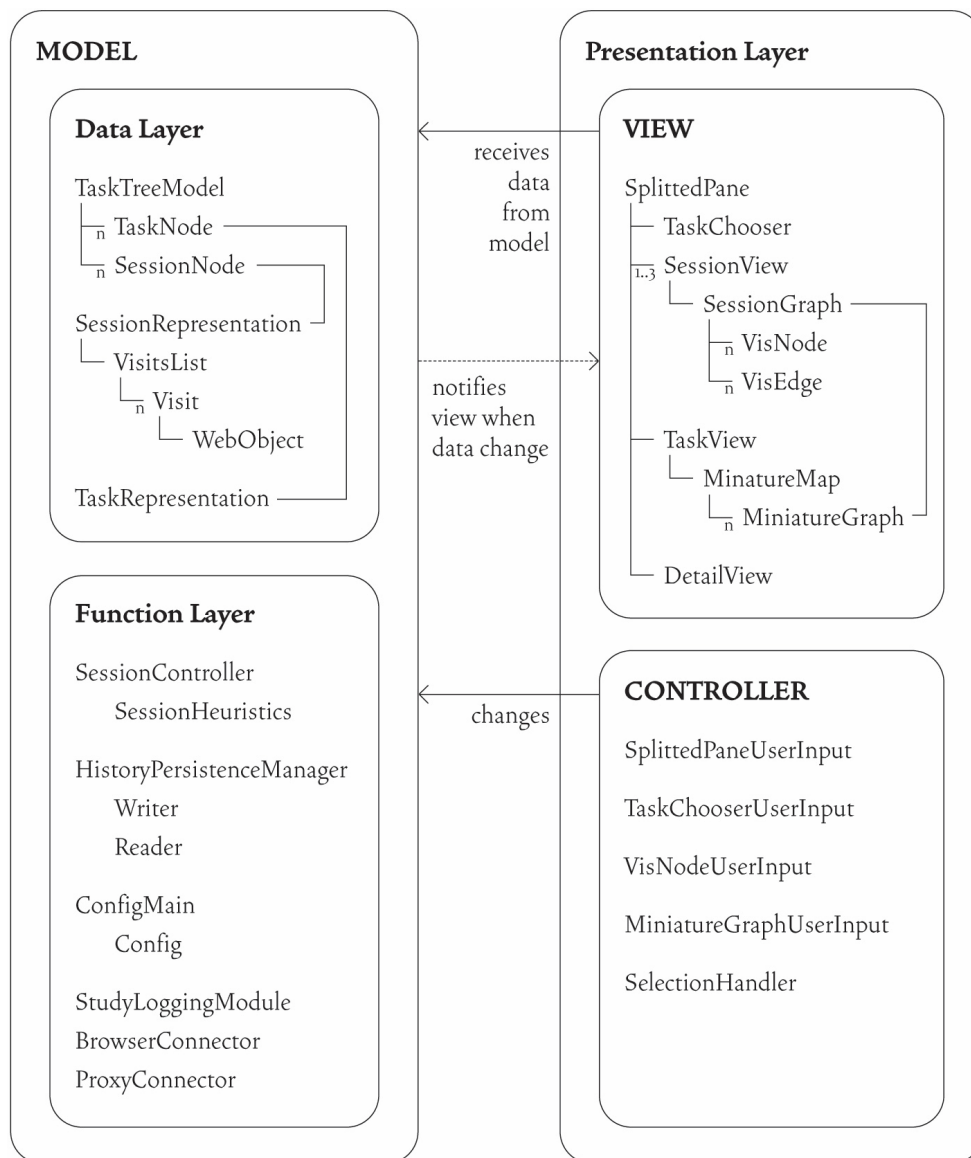


Figure 3-19: The prototype's internal architecture.
 The structural solution is guided by the combination of the Model-View-Controller pattern and the 3-tier architecture (data-function-presentation).

The *presentation layer* comprises the user interface, divided into user input (**CONTROLLER**) and system output (**VIEW**). A strength of the MVC pattern is that it decouples data from their appearance. While the **MODEL** contains the core functionality and data independent of specific output representations or input behavior, the **CONTROLLER** is used to modify these data and access related functionality. The **VIEW** obtains data for presentation from the underlying **MODEL** and is updated whenever the **MODEL** changes. A **VIEW** consists of a **SplittedPane** that presents the GUI as shown in Figure 3-4. The **TaskChooser** provides an interactive view of the **TaskTreeModel**. Up to three **SessionViews** are provided in tabs to show the session

visualizations. They reference **SessionGraphs**, which represent the user's path using nodes (**VisNodes**) and edges (**VisEdges**). The **TaskView** shows the spatial arrangement of miniature versions of the referenced session graphs (**MiniatureGraphs**), the **MiniatureMap**. The **DetailView** shows details on demand, including thumbnail, title and URL. The **SplittedPane** itself is responsible for providing the menu and toolbar at the top.

The **CONTROLLER**, also part of the *presentation layer*, handles user input on the **SplittedPane**, i.e. menu selections, keyboard shortcuts, and tab interaction. Special classes handle user interactions with the **TaskChooser** like the dragging and dropping of SG-session nodes to different locations within the SG-task hierarchy. Other classes handle interaction with single **VisNodes** embedded in the **SessionGraphs**, and with **MiniatureGraphs**. A **SelectionHandler** deals with the selection of one or multiple user interface objects. These classes prompt the **MODEL** to change the underlying data or to execute specific functionality. The facade pattern (Gamma et al. 1996: 189) is used to define one specific access point in the Function Layer for all distributed controllers that are interested in using it.

3.3.2 Implementation

The above-described architectural and software-technical design decisions resulted in an implementation that is described subsequently.

3.3.2.1 Static Structure - Packages and Important Classes

The final version of prototype A – as used for the longitudinal study – was implemented in Java 1.4.1. It consists of 147 classes and is organized in eight different packages. It comprises 16, 203 source lines of code. Table 3-6 gives an overview on the most important classes and packages. The abbreviation **sga** stands for Session-Graphs approach. An overview for all classes is given in appendix 9.3.1.

The package **sga.model** comprises the core classes of essential functionality and the models of domain data. **SGASconePlugin** connects to Scone and is instantiated on system start. **SessionController** controls most of the application's functionality. **CmdTable** acts as a facade to controller classes (Gamma et al. 1996: chapter 4) and distributes their calls to appropriate receivers. **WebObject** and **Visit** are models of web pages with fields such as URI and title and the related visits (see Figure 3-20 for an overview of fields and methods).

Package **sga.ui** is responsible for the non-Jazz parts of the graphical user interface (GUI). The **SplittedPaneFrame**, a **JFrame**, holds the complete GUI which is divided into the five sections as described above (compare Figure 3-4). The **TaskChooser**, a **JPanel**, presents the hierarchical view of SG-tasks. **SessionView** and

TaskView, both **JPanels**, hold the Jazz visualizations (**ZCanvases**). Several utility classes are located in this package to support GUI interaction and the drag and drop manipulation of tree nodes in the task chooser.

The package **sga.sessions** holds classes that model SG-sessions and their related behavior. Instances of **SessionRepresentation** model the single SG-sessions. **DefaultSessionHeuristics** is used to determine the session borders (see page 116). It implements the interface **SessionHeuristicsIF** that is exclusively accessed by **SessionController**. Thus, it can be easily replaced in order to study the effects of alternative system behavior. The sub-package **sga.sessions.sessiongraphs** contains all Jazz related classes for SG-session and SG-task visualization. **SessionGraph** extends **ZLayerGroup** while **VisNode** and **VisEdge** extend **ZGroup**. Two positioner classes are responsible for initially placing a new node in either a task or a session view; other classes manage the animation of the views over time. **MiniatureMap**, a **ZLayerGroup**, and **MiniatureGraph**, a **ZNode**, are used to model task views and the single miniature graphs inside. In addition, the package contains **SGCamera** and **MiniatureMapCamera**, two **ZCameras** that are marker classes for identification of the current view type (session or task view) during rendering.

sga.tasks contains **TaskRepresentation** that models SG-tasks. The data model for the SG-task tree is implemented in **TaskTreeModel**, a **DefaultTreeModel**. Two kinds of tree nodes are defined, **SessionNode** and **TaskNode**, as well as a common superclass, **TaskTreeNode** (a **javax.swing.tree.DefaultMutableTreeNode**) that is used to organize SG-sessions and SG-tasks within the tree.

sga.io provides the **HistoryManager** that stores the visual history. It utilizes a specific writer and reader class. An interface facilitates different output formats (e.g. single versus multiple file output). The final version of prototype A uses the two classes **SGAFileWriterMultiFiles** and **SGAFileReaderMultiFiles**.

sga.log is responsible for logging the participants' activities during the longitudinal study. Most of the work is accomplished by **AutoLogger**, which maintains the two files in which overview and detail data are stored for study purposes.

sga.config comprises classes responsible for managing user-defined settings (colors, node sizes, system behavior, etc.). The user can create several sets of settings, label them, save them, and reuse them. All views immediately react when settings are changed. **ConfigMain** manages the configuration handling and controls related file readers and writers to store the settings persistently. Several classes handle the modifiable values for numbers, boolean data, and colors – e.g. **ColorObject**. They facilitate the simple modification of values of objects already instantiated. These classes are used by GUI templates – like **ColorTemplate** – in the preference dialog, e.g. to model the values of range sliders and check boxes. The GUI itself is determined by the classes **ConfigWindow**, **ConfigContentPanel** and specific template classes such as **ConfigContentSliderTemplate** for sliders or **ConfigContentColorTemplate** for the presentation and modification of color values.

sga.model	Models the domain context
SGASconePlugin	Connection to Scone
SessionController	History and GUI related functionality
CmdTable	Facade for controlling classes
WebObject	Can model any visited web page
Visit	Models an actual visit
sga.ui	Provides the GUI
SplittedPaneFrame	Main GUI – holds views (TCH, SV, TV, DV) & menu
TaskChooser	Hierarchy of SG-sessions and SG-tasks
SessionView	Can hold a visual representation of an SG-session
TaskView	Can hold a visual representation of an SG-task
sga.sessions	Models sessions and their visualizations
SessionRepresentation	Represents an SG-session – structurally & visually
DefaultSessionHeuristics	Determines SG-session borders
sga.sessions.sessiongraphs	Jazz visualizations of tasks and sessions
SessionGraph	Jazz visualization of an SG-session
VisNode	A visual node representing a web page
VisEdge	A visual edge, representing one step of a trajectory
MiniatureMap	Jazz visualization of an SG-task
MiniatureGraph	Miniature of an SG-session used by the Min.-Map
sga.tasks	Models tasks and their hierarchy
TaskRepresentation	Represents an SG-task – structurally and visually
TaskTreeModel	Models the hierarchical structure of SG-tasks
SessionNode	SG-session container for the TaskTreeModel
TaskNode	SG-task container for the TaskTreeModel
sga.io	Manages all in- and output
HistoryManager	Manages history persistency
SGAFileWriterMultiFiles	Used to store the history and visualizations
sga.log	For logging during longitudinal study
AutoLogger	Reads and writes study logs
sga.config	Manages user adaptable settings
ConfigMain	Encapsulates functionality for configuration
ColorObject (and others)	A user-definable color
ColorTemplate (and others)	GUI element to define color
ConfigWindow	GUI of the Settings dialog
ConfigContentSliderTemplate	Models GUI sliders
sga.utils	Mixed helper classes
DateUtils	Manages date-based calculations
BrowserControl	Used to control the browser
SessionGraphsEvaluator	Used for the controlled experiment
ThumbnailGenerator	Generates thumbnails as parallel threads

Table 3-6: SessionGraphs implementation - packages and selected classes.

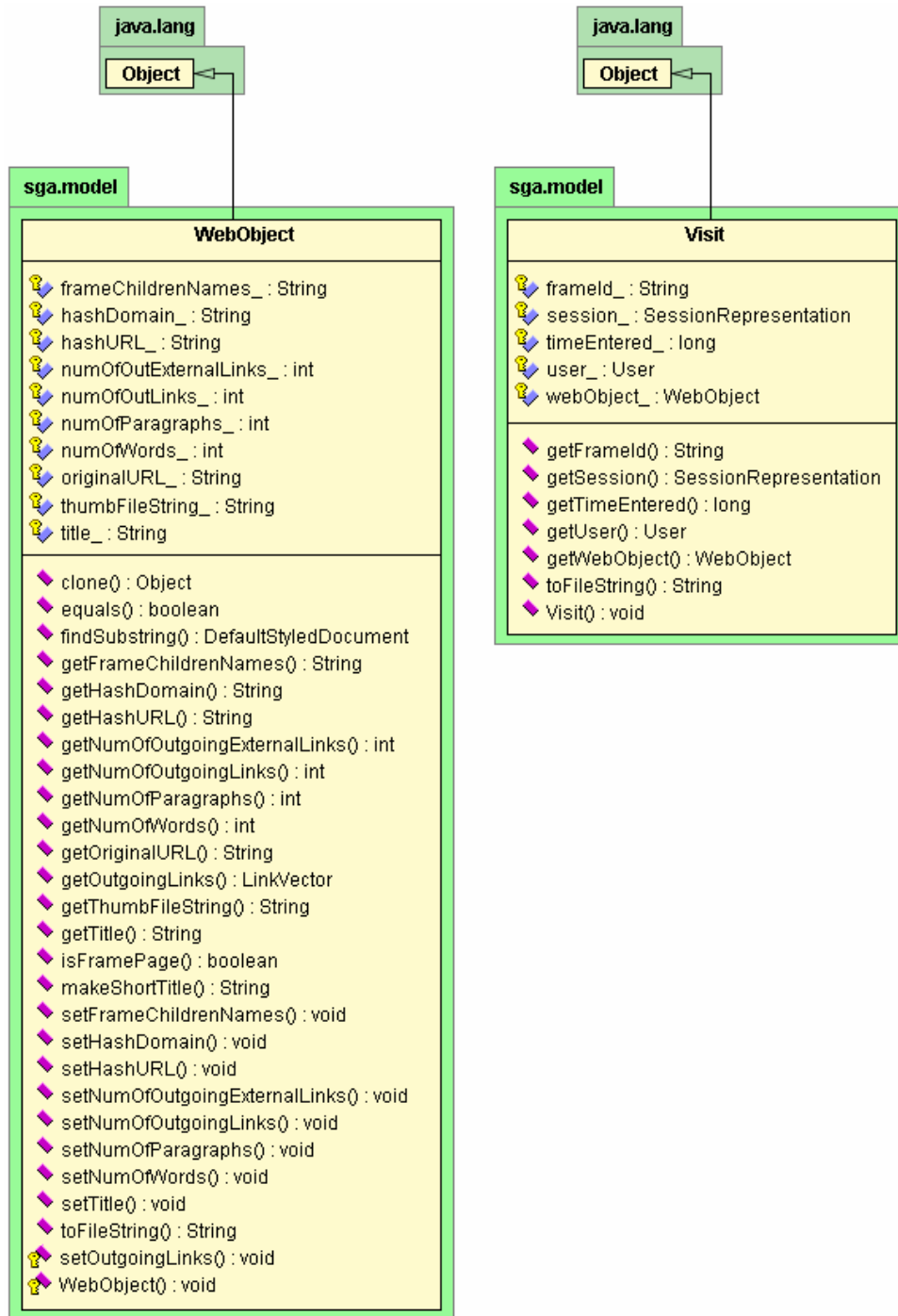


Figure 3-20: Principal fields and methods of *WebObject* and *Visit*.

`sga.utils` contains helper classes like `DateUtils` for date related calculations, the modified `BrowserControl` class to control the web browser, the class `SessionGraphsEvaluator` used in the controlled study, the `ThumbnailGenerator` class, as well as several print utility classes for development and debugging purposes.

While the previous discussion delineated the static structure of the prototype, the following describes how dynamic behavior is handled.

3.3.2.2 Dynamic Behavior

The SessionGraphs prototype is in charge of a variety of responsibilities that will be elaborated below:

- To receive and process incoming events.
- To determine the SG-session and SG-task that a visit belongs to.
- To set up the hierarchical organization.
- To visualize SG-sessions and SG-tasks appropriately.
- To process user interactions.
- To enable persistency for history, visualization, and study data.

Processing Incoming Web Browser Events

For any visited web page, Scone sends three different **AccessEvents** to the **SGA-SconePlugin**: One when the page starts loading, one when its source is completely received, and one when the page is unloaded by the browser (see Figure 3-21).

The first event tells the plugin that the web browser has just started to load a new web page. This may be an HTML page – modeled by **scone.netobjects.-HtmlNode** – or any other object – modeled by **scone.netobjects.NetNode**. The event is passed to the plugin's **update** method, which internally creates a new **WebObject** that stores all information about the visited node that is already accessible at this point in time (URL and title). In addition, a **Visit** object is created that collects data referring to the specific visit, but not to the object itself – at this point in time only the start date and a reference to the **WebObject**.

The plugin then calls **SessionController's newVisitOccurred** method with the **Visit** as parameter. **SessionController** determines the SG-session to which this **Visit** belongs by calling **DefaultSessionHeuristics.determineSession** (see page 116). If the web page was already visited during the current SG-session, the corresponding **VisNode** will be highlighted as the “current one”. Otherwise a new **VisNode** is added to the visualization. In a parallel thread **ThumbnailGenerator** creates a thumbnail. References to **WebObject** and **Visit** objects are temporarily stored to enrich them with more detailed information as soon as subsequent events arrive from Scone.

The second Scone event is created immediately after the web page has been completely parsed and loaded into the browser, not including its embedded media. This may take several seconds or even minutes after the first event, depending on the page size and transmission rate. To satisfy response time requirements, this potential delay demands that visits are already visualized after the first Scone event – even if the data are still incomplete.

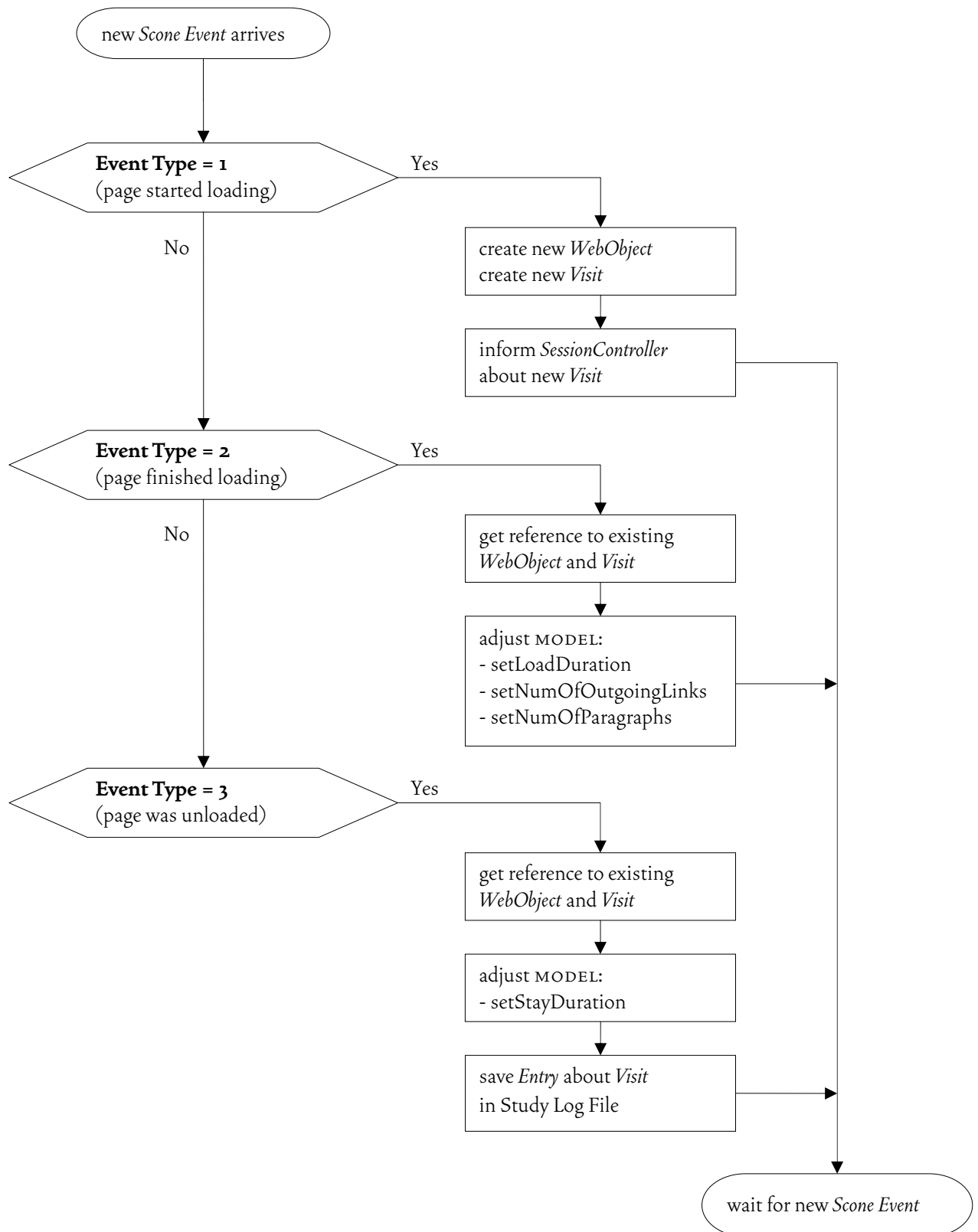


Figure 3-21: Events after a page visit.

Each web page visit fires three Scone AccessEvents that evoke different actions.

After the second event is received, the **Plugin** uses the temporary references to fill **WebObject** and **Visit** objects with the now available data (number of words, paragraphs, outgoing links, etc.). This may also influence the views – for example, the visual representation of **VisNodes** reflects the number of outgoing links by showing zero, three, six, or twelve small round satellites grouped around a page representation (similar to the WebNet project).

The third event is fired immediately after the user leaves a page. This can happen if he or she replaces it with another page in the same browser window or after the window is closed. Then the staytime is calculated and added to the **Visit** object. The class **AutoLogger** exports the visit information to the detailed log file for study purposes.

The *current* SG-session is set to the SG-session returned by the **determineSession** method as mentioned above. The *current* SG-task is adjusted to the appropriate SG-task and all views are updated accordingly. Finally, **SessionController** tells the current SG-session about the new visit via the **setCurrentPage** method.

The described procedure proved to be a straightforward way for implementing a session-based history visualization. It is implemented in a structured, easily modifiable way, which was important for quickly adjusting the system's behavior throughout the development period, especially after feedback from study participants and colleagues.

Determination of Session and Task

For each visit, the prototype has to determine to which SG-session it belongs. For this purpose, **SessionController's processNewAccess** method calls the method **determineSession** in **DefaultSessionHeuristics**. Based on three parameters – current SG-session, window identifier, and load method – **determineSession** returns either a reference to an existing SG-session or creates a new one.

The applied heuristic operates as follows. Each new visit is automatically added to the current SG-session, except in the following situations. (1) When the prototype launches, a new SG-session is created, (2) whenever the current SG-session is null – e.g. after deletion of SG-sessions in the task chooser and no SG-sessions remain, (3) whenever the user actively requests the start of a new SG-session using the new session command, or (4) after the user requests the function “search the web” with the preferred search engine.

The system switches to another existing SG-session whenever the visit occurs in a window containing a windowID that is currently open which belongs to another SG-session. Then the current SG-session is replaced by this SG-session.

New SG-sessions can be manually created using the new session command, e.g. via a task bar button, the menu bar, a keyboard shortcut, or by selecting the context menu on a **VisNode**. In the last case, a function “open a new session with this node” enables the user to copy the selected node into a new SG-session and then

continue with the existing node in this new SG-session. Alternatively, the entry “cut this node” enables the user to delete a selected node from an existing SG-session and paste it into a new one.

Visualization of Sessions

To visualize the three most recently viewed SG-sessions, a **JTabbedPane** is used which can hold up to three **SessionViews**, each of which extends **JPanel** and contains a Jazz visualization of the related SG-session. A **SessionView** object is created by calling the **showSessionView** method of **SessionController** with the desired **SessionRepresentation** as a parameter.

Whenever the system tells the **SessionView** to display a certain SG-session, the **TaskView** also updates its view, showing miniature versions of all SG-sessions that belong to the current SG-session’s SG-task – several **MiniatureGraph** objects spatially arranged in one **MiniatureMap**. The current SG-session’s miniature graph is highlighted in the task view by a rectangular outline. The session graph in a session view is rendered on a Jazz **ZCanvas**.

Figure 3-22 shows an exemplary underlying Jazz scenegraph structure, defining a small session graph with two **VisNodes** and one connecting **VisEdge**. The **SessionGraph** itself is added to the scene graph in line 3. It extends the **ZLayerGroup** so that a **ZCamera** can view it. The **SessionGraph** object defines both the visualization for the **SessionView** and the **TaskView**, depending on which camera is looking on it. It contains an **SGLabel** (a **ZGroup**) which displays the SG-session’s title next to the miniature graph in the **TaskView** (lines 4 to 6). The two **VisNodes** are attached to **ZTransformGroups** that define their 2D position (line 7 and 18). As shown for the first **VisNode**, it contains two different representations that are rendered either in a **SessionView** or in a **TaskView**. Lines 9 to 14 define the detailed representation for the **SessionView** comprising both an ellipse and a title, whereas lines 15 to 17 define only the pure ellipse representation for the **TaskView**. Formally, a **VisNode** is a 6-tuple consisting of the referenced web page, a title that will be used in the visualization, and x- and y-values that specify a position in 2D space, a size, and a color value. The **VisEdge** is represented once for each of the two view types, both times as a **ZPolyline**, but with different pen widths (lines 19 to 25).

Figure 3-23 shows how the render method of class **VisNode** dynamically selects the appropriate visual representation either for the **Task-** or the **SessionView**. The two visual representations are created in the constructor of **VisNode** (line 5 and 6), and then added to the scene graph (line 7 and 8, resulting in lines 9-14 and 15-17 of Figure 3-22). Based on the class name of the rendering camera (lines 16-22), the method chooses the appropriate visualization (line 23) and renders it (line 24). **SGCamera** and **MiniatureMapCamera** (both **ZCameras**) are pure marker classes that contain no code themselves.

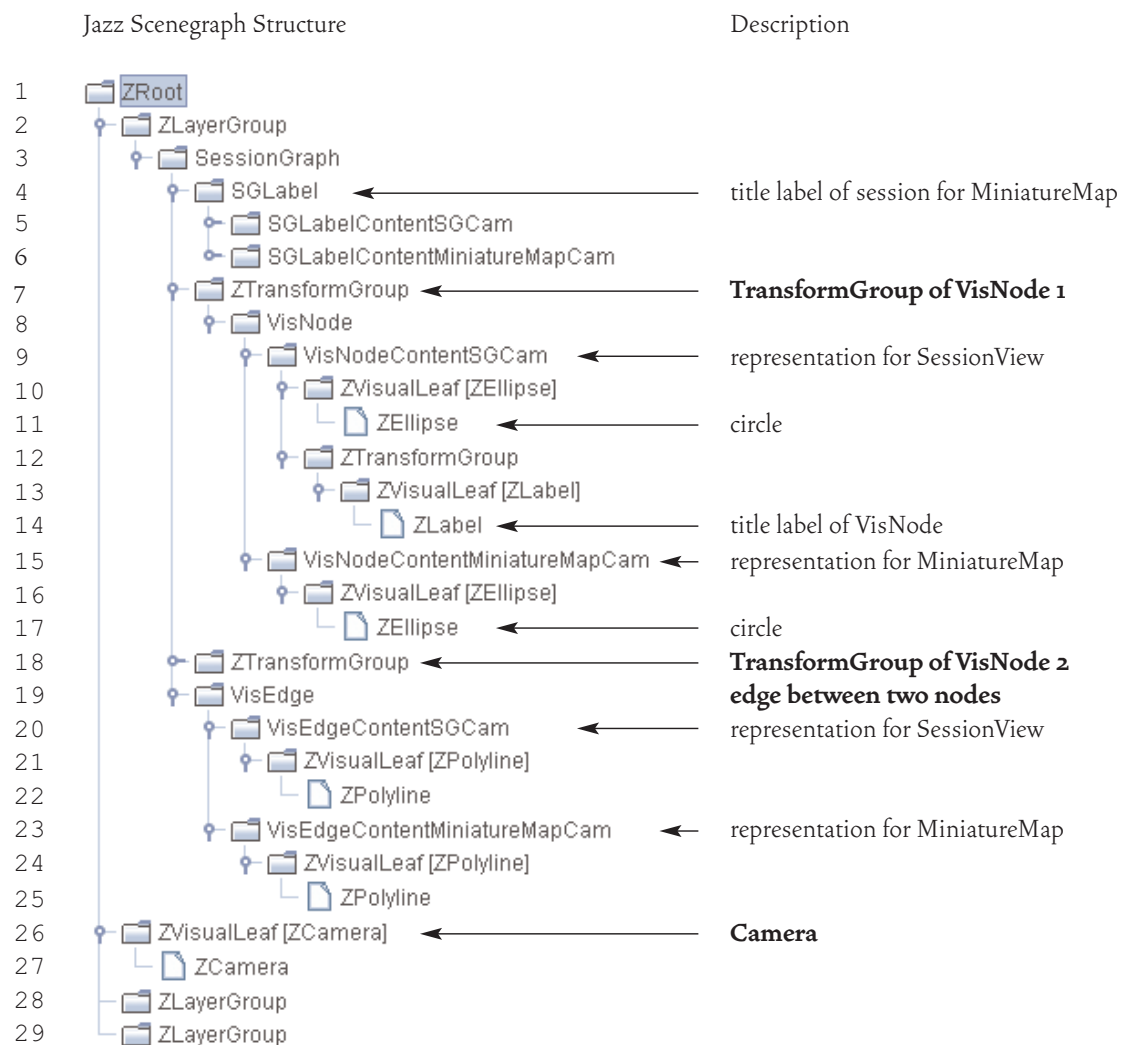


Figure 3-22: A SessionView's internal Jazz scenegraph structure. A simple example with two nodes and one edge.

In order to enable scrolling, the **ZCanvas** is integrated into the **SessionView** via a **ZScrollPane** that allows scrolling the session view once it grows larger than the visible area.

Animating the Session Visualization

A *force-directed placement algorithm* creates the behavior responsible for the impression of a fluid surface. The illusion of continuous motion is created by many subsequent discrete steps of animation. For this purpose the method **relax** in **StoppingLiquidity-NodePositioner** is called every 20 milliseconds. The subsequent execution of the underlying placement algorithm causes successive small position changes to all moveable nodes.

The underlying force-directed placement algorithm models both attractive and repelling forces of a physical spring system based on the related session graph.

```

1 private VisNodeContentIF[] twoRepresentations_ = new VisNodeContentIF[2];
2
3 public VisNode(WebObject webObject, ...) {
4     ...
5     this.twoRepresentations_[0] = new VisNodeContentSGCam(this);
6     this.twoRepresentations_[1] = new VisNodeContentMiniatureMapCam(this);
7     this.addChild((ZTransformGroup)bgCamContent_);
8     this.addChild((ZTransformGroup)bimCamContent_);
9     ...
10 }
11
12 public void render(ZRenderContext renderContext) {
13     int currentlyViewedIn;
14     ZCamera currentCamera = renderContext.getRenderingCamera();
15
16     if (currentCamera.getClass().getName().equals(
17         "sga.sessions.sessiongraphs.SGCamera")) {
18         currentlyViewedIn = 0;
19     } else if (currentCamera.getClass().getName().equals(
20         "sga.sessions.sessiongraphs.MinatureMapCamera")) {
21         currentlyViewedIn = 1;
22     }
23     VisNodeContentIF visNodeContentIF = twoRepresentations_[currentlyViewedIn];
24     visNodeContentIF.render(renderContext);
25 }

```

Figure 3-23: Code that renders either the session- or the task view. The appropriate visual representation is chosen in the render method of class `VisNode`.

It is a modified version of the spring-based algorithm presented by (Eades 1984) that ships with the Java SDK («GraphLayout demo», Sun 2003). The graph's edges are interpreted as springs and its nodes as rings that connect the springs. The algorithm oscillates until it finds a stable configuration, using a minimal energy function, in which attractive and repelling forces are balanced. Figure 3-24 shows a pseudo-code representation of the applied algorithm.

The algorithm is divided into three parts. Part one iterates over all edges and adds a force vector to each of the two nodes connected by the edge (lines 6 to 15). This leads either to a repelling or an attractive impulse based on the deviation of this edge's length from an ideal edge length. The higher the deviation, the larger the force. The relative position change within one animation step – and thus the perceived speed of movement – increases with this force. This behavior is similar to a physical spring system. The calculated force vector is stored in a variable $\vec{\delta}$ per node. Each node may be involved in several edges and all related forces are accumulated in this node's $\vec{\delta}$.

In the controlled experiments, users gave the feedback that the fluid behavior of the layout was a good idea – they liked the vivid impression. The nodes, however, did not stop moving in this version of the prototype. This was considered a drawback. According to users' feedback the nodes should move initially, but quickly begin "freezing" until they no longer move, so that the session visualization can be laid out manually without the algorithm changing a node's position again.

```

01 StoppingLiquidityNodePositioner.relax() {
02   Given: Session Graph  $g = (N, E)$ . Initialize each node's
03   vector  $\vec{\delta}$  to accumulate its spatial change in this relax-
04   ation step before actual movement:  $\forall \vec{n} \in N: \vec{n} \cdot \vec{\delta} = \mathbf{0}$ 
05   Part 1) Calculate forces between adjacent nodes
06    $\forall \vec{e} \in E \{$ 
07     Determine for all edges a force  $\vec{f}$  based on the
08     difference between the current and an ideal edge length.
09      $coefficient = (idealEdgeLength - |\vec{e}|) / (3 \cdot |\vec{e}|)$ 
10      $\vec{f} = coefficient \cdot \vec{e}$ 
11     Adjust the  $\vec{\delta}$  of the edge's start and destination node.
12      $\vec{e}.dest.\vec{\delta} = \vec{e}.dest.\vec{\delta} + \vec{f}$ 
13      $\vec{e}.start.\vec{\delta} = \vec{e}.start.\vec{\delta} - \vec{f}$ 
14   }
15   Part 2) Prevent overlapping nodes by repelling forces
16    $\forall \vec{m} \in N \{$ 
17     Introduce a cumulative repelling force  $\vec{\theta}$  per node.
18      $\vec{\theta} = \mathbf{0}$ 
19      $\forall \vec{n} \in N, \vec{n} \neq \vec{m} \{$ 
20       square distance  $d^2 = |\vec{n} - \vec{m}|^2$ 
21       if ( $d = 0$ ) {
22         Add a random vector  $\vec{\rho}$  with  $-1 \leq \vec{\rho}_{x,y} \leq 1$ .
23          $\vec{\theta} = \vec{\theta} + \vec{\rho}$ 
24       } else if ( $d < 100$ ) {
25         Increase the accumulative repelling force by an
26         amount that decreases with growing distance.
27          $\vec{\theta} = \vec{\theta} + \left( \frac{\vec{n} - \vec{m}}{d^2} \right)$ 
28       }
29     } Add the accumulated repelling force to each node's  $\vec{\delta}$ .
30      $\vec{m}.\vec{\delta} = \vec{m}.\vec{\delta} + \frac{\vec{\theta}}{|\vec{\theta}|}$ 
31   }
32   Part 3) Reposition each node to its new position
33    $\forall \vec{n} \in N \{$ 
34      $\vec{n} = \vec{n} + \vec{n}.getCurrentGlueValue \cdot \vec{n}.\vec{\delta}$ 
35      $\vec{n}.\vec{\delta} = \frac{\vec{n}.\vec{\delta}}{2}$ 
36   } Repaint SessionView and TaskView.
37   If maximum position change over all nodes is below 0.7:
   apply frozen state, stop animation, do not call relax
   again.
}

```

Figure 3-24: Force directed placement algorithm for animation.

Used to support the *fluid surface metaphor* derived from video analysis.

Satisfying their request would also diminish another potential problem. If the graphs would always be “fluid”, i.e. if its nodes were to be permanently repositioned according to the automated force-directed placement, the graph would often change its visual shape, reducing the users’ ability to recognize a graph based on this characteristic. Automated or manual re-layout would change all nodes’ positions and the lengths and slopes of the edges, even though the topology would remain unchanged; absolute and relative positions would change and characteristic visual constellations would vanish. Misue et al. (1995) showed that changing the layout of a graph severely compromises people’s ability to remember it, even when the graph’s underlying topology does not change. Letting the nodes “freeze” and keep their positions after an initial time of movability would thus contribute to a much more stable shape consequently increasing its recognizability.

Thus, the algorithm was modified so that immediately after addition, a node’s movability is maximal. Then the movability slowly decreases until, after about one minute, the node becomes “frozen” at its current position. Compliant to the water surface metaphor, this behavior creates the illusion as if the node would become heavier, or start to touch the “ground”, until it finally cannot move anymore.

The advantage of this method is that the user can exactly manipulate the position of the frozen nodes without worrying that the algorithm will modify their position again. The time span of one minute allows the force-directed placement algorithm to smoothly situate the new node in a position where the forces are highly balanced, such that freezing the view does not stop the animation abruptly or impede balanced layouts.

The method `relax` in `StoppingLiquidityNodePositioner` is called repetitively to calculate the single phases of the session graph’s animation.

The second part of the algorithm (lines 15 to 31) prevents mutual overlap between the nodes. Repelling forces are applied between two nodes if their distance drops below a certain threshold. This is accomplished by comparing each node to all the other nodes. The repelling force is greater when the nodes are closer together (line 27). For the case when the distance is zero – i.e. two nodes are located in exactly the same position – a random force vector is applied (line 23). Line 30 adds the calculated force to the node’s current $\vec{\delta}$.

Before executing each node’s position change, its “fluidity” has to be calculated in order to simulate the process of slow freezing within one minute after insertion. This is accomplished in part three (lines 32 to 35) by applying a “liquidity factor” to each node. The method `getCurrentGlueValue` in `VisNode` returns the node’s current “liquidity”. It is 1.0 at the node’s creation time (highest movability, maximum animation) and 0.0 one minute after creation time (no movability, no animation). Each node’s liquidity factor is then multiplied with the accumulated force $\vec{\delta}$ of each node as determined in the two precedent steps (line 34). This procedure induces the desired slowly reduced liquidity. Nodes older than one minute show no

liquidity and thus no movement; applied forces have no effect. As requested by several users, the possibility is offered to manually reset the liquidity of all nodes to 1.0, using a menu command, in order to regain maximum movement and to automatically reset the view. Then, the 60 seconds counter starts over and slowly decreases the movability of all nodes again.

The **relax** method is called over and over again until all position changes fall under a certain threshold. Then the animation stops. Animation will be restarted by the **SessionController** after each interaction with the visualization, e.g. after manual movement of a node. The **SessionController** will activate the positioner via the class **SGAnimator**. Its method **animateOneTime** (see

Figure 3-25, part one) calls the positioner's **relax** method for one relaxation step and forces the thread to sleep for 20 milliseconds before calling itself again to regulate the speed of the animation.

Figure 3-26 gives an impression of the animated behavior of a session graph, showing six subsequent temporal miniatures. Miniature a) to c) show how a new web page (labeled "G. Park F") is added to the existing graph. Miniature b) shows the entrance of the new node. At this moment, attractive forces are attached to the new node and it is smoothly attracted by node "G. Travel". It moves until a predefined optimal edge length is reached in c). Miniatures d) to f) show how the still liquid graph behaves on mouse interaction. The node "Volunteer G." is dragged in d) and e) and the remaining nodes slowly follow until the underlying forces are again balanced – as depicted in miniature f).

Placing new Nodes

The resulting layout is highly influenced by the strategy according to which the initial position of a newly inserted node is determined. For example, it makes a big difference whether a new node is placed in the center of an existing, relatively large

```

1 // In class SGAnimator:
2 public void animateOneTime() {
3     if (ConfigMain.getCurrentConfig().ANIMATE_VIEWS
4         && !SessionController.isFrozen_) {
5         SessionController.getCurrentSession().getSessionGraph(
6             .getPositioner().relax();
7         Thread.sleep(20);
8         SwingUtilities.invokeLater(new Runnable() {
9             public void run() {
10                SGAnimator.this.animateOneTime();
11            }
12        });
13    }
14 }

```

Figure 3-25: Code that triggers the single animation loops.

The view of the current **sg-session** is animated by repetitively calling the **relax** method of the related positioner every 20 milliseconds.

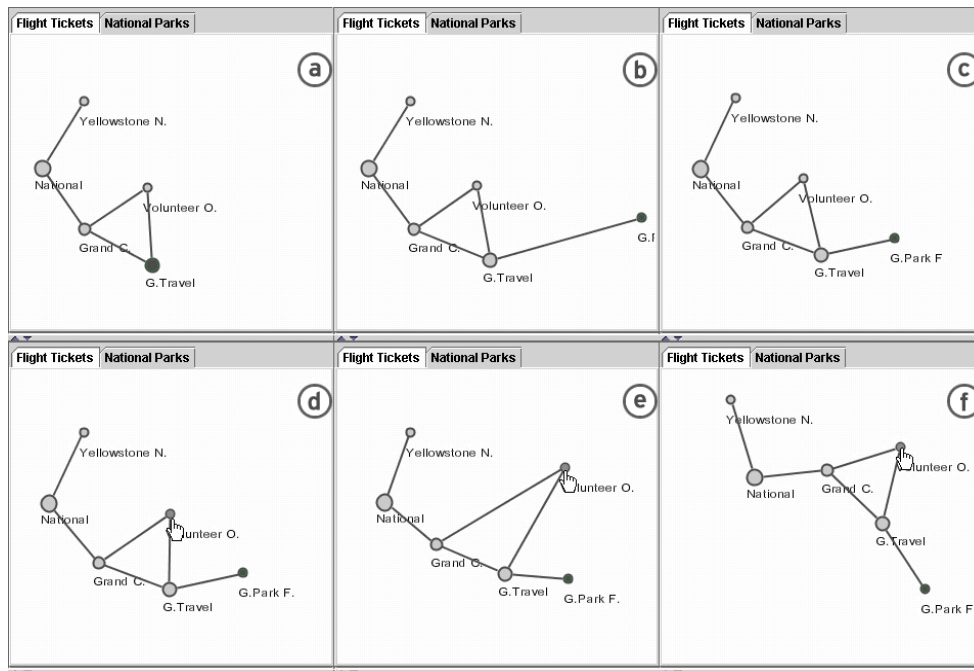


Figure 3-26: Smooth animation using the force-directed algorithm.

- a-c) A new web page is dynamically attached to the current one from the right.
d-f) The user manually reshapes the graph that moves smoothly as reaction.

session graph or at its border. Furthermore, the dynamic forces have to be considered that will affect the new node immediately after it appears as described above.

In order to examine different placement procedures, the interface **PositionerIF** was created, which defines besides its **relax** method for animation a **place** method responsible for placing a new node. The finally used implementations are **StoppingLiquidityNodePositioner** for **relax** and its superclass **StoppingNodePositioner** for **place**.

The design of the placement algorithm was guided by three precepts. First, the new node should be visually perceivable by its motion for a reasonable amount of time. This meant that the new node had to be placed so that reasonably high forces were attached that make it move to its final position, i.e. it should be inserted far enough from its final position. Second, a new node – if not the first in a session graph – should be connected to this graph’s “current node” by an edge in order to maintain the SG-session’s structure. Third, edge crossings should be minimized for better readability of the view. In case a view already contained several nodes and the current node was located on its right edge it would not be advisable to place a new node in the middle of all nodes or even beyond the left edge since many edge crossings could be the consequence – even with the force-directed placement algorithm doing its work.

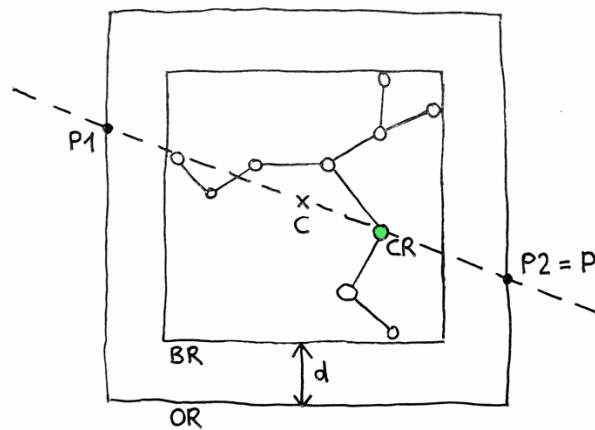


Figure 3-27: Placing a new node in a session graph at P2. This method helps to reduce edge crossings.

These demands led to the development of an algorithm working as follows. It considers where the current node is located in relation to the remainder of nodes in the session visualization. Then it chooses a position for the new node that reduces possible edge and object crossings in a five step process (see Figure 3-27).

- 1) Determine the center C of the bounding rectangle BR of the existing session graph.
- 2) Determine an outer rectangle OR that surrounds the bounding rectangle of the session graph on each side with the same predefined distance d.
- 3) Determine the two intersection points P1, P2 of OR and a straight line defined by C and the *current node* CR.
- 4) Choose the intersection point P (either P1 or P2) that is closer to the currently active node CR or P1 if both distances are equal.
- 5) Place the new node at P.

Two special cases are treated separately. When the new node is the first node in an SG-session it is placed in the center of the view. When the new node is the second node in an SG-session it is placed at a certain predefined distance and angle aside the first node. The angle is determined in combination with a random factor so that any two session visualizations will not look the same.

Visualization of Tasks

An SG-task – as introduced in chapter 2.2.3.1 – consists of 0 to n SG-sessions and of 0 to n sub-tasks. The chosen concept of visualizing SG-tasks is simple. A visual overview is given over all SG-sessions that are direct children of this SG-task. The task view shows a miniature map that comprises miniaturized versions of all session graphs that belong to the related SG-task. SG-sessions that are nested in sub-tasks and the sub-tasks themselves are omitted in the a task view.

Miniature graphs (see Figure 3-4-D) reduce session graphs to nodes and edges, hiding the node titles. In the miniature map SG-session titles are shown when the mouse hovers over a miniature graph.

Miniature graphs show exactly the same shape like session graphs only reduced in size. Node colors and even movement during interaction with a session graph or while the animation is still going on is immediately reflected in the corresponding miniature graph. This behavior is technically enabled by referencing the corresponding session graph (Figure 3-28, line 2 to 5). In its **render** method, **MiniatureGraph** simply loops through the call to the render method of the corresponding session graph (line 17) – after drawing a bounding rectangle around it so that this miniature graph can be perceived as the current SG-session in the miniature map (lines 12 to 16). In the render method of the session graph (see Figure 3-23, line 12 to 25) the miniature map representation is chosen dynamically based on the **MiniatureMapCamera** marker.

The single miniature graphs in a miniature map are animated similarly to the nodes in a session graph; however, here miniature graphs are treated by the algo-

```

1 public class MiniatureGraph extends ZNode implements IconListenerIF {
2     private SessionGraph sessionGraph_;
3
4     public MiniatureGraph(SessionGraph sg) {
5         this.sessionGraph_=sg;
6         ...
7     }
8
9     public void render(ZRenderContext aRenderContext) {
10        Graphics2D aG2 = aRenderContext.getGraphics2D();
11
12        if (isCurrent_) {
13            aG2.setColor(ConfigMain.getCurrentConfig().NODE_BORDER_COLOR.getColor());
14            aG2.setStroke(new BasicStroke(6));
15            aG2.draw(getBoundsReference());
16        }
17        sessionGraph_.render(aRenderContext);
18    }
19
20    protected void computeBounds() {
21        getBoundsReference().setRect(sessionGraph_.getBounds());
22    }
23
24    ...
25 }

```

Figure 3-28: Code that renders miniature graphs.

rithm like the single, animated nodes in a session graph. New miniature graphs are inserted into the plane of existing graphs using the `position` method in `SmoothMiniatureMapPositioner`. They are always placed at the same predefined position and immediately repositioned in smooth animation in case of overlapping with existing miniature graphs. Animation is controlled by the `relax` method in `SmoothMiniatureMapPositioner` and is started whenever the user moves a miniature graph manually or a new graph is inserted. It works similarly to step 2 of the `relax` method applied to session graphs that is responsible to prevent the overlapping of nodes. Repelling forces are added between any two miniature graphs to keep a minimum distance between them. The visualization appears similarly “fluid” or “liquid” as that of the session view and keeps the GUI perceptually coherent – it stays conform to the introduced visual metaphor of smooth, fluid animation.

Dealing with User Interactions

Session and task views are implemented in Jazz. This enables interface events (`ZMouseEvent`s) on any desired level. An event can be handled e.g. on the `VisNode` level if the result only addresses this node or on the session graph level if it relates to the entire graph. Several mouse events on miniature graphs (mouse clicked, pressed, released, and exited) are directly passed through to the related session graph to modify its view accordingly (e.g. setting or removing a single node’s highlight that indicates “mouse is over”).

A `VisNodeMouseAdapter` is attached to each `VisNode` that is responsible for handling mouse events. When the mouse enters a `VisNode`, a highlight is set indicating that the mouse is currently above this node. Detail information including a thumbnail of the page is displayed in the detail view. When the mouse exits, the highlight and detail information are removed.

As the mouse is clicked on a `VisNode`, its corresponding URL is retained and the referenced page is loaded into the web browser. This is accomplished either by the applet connector or the `BrowserControl` class (compare page 107).

Jazz also supports many interaction features. The class `SGCompositeSelectionHandler` (a `ZCompositeSelectionHandler`) enables the programmer to comfortably initialize several event handlers at once: the `SGSelectionModifyHandler` that extends `ZSelectionModifyHandler` to control the selection of `VisNodes` (e.g. by applying control keys for selection of multiple objects), an `SGSelectionMoveHandler` (extending `ZSelectionMoveHandler`) to control the movement of selected items, as well as a `ZSelectionDeleteHandler`, a `ZSelectionScaleHandler`, and a `ZSelectionResizeHandler`. Thus, the user can select single or multiple nodes at once and can cut them or move them around.

Handling Frame Sets

The aforementioned description prescinds from the treatment of frame pages as used by HTML 4.0 which builds a special technical challenge. The frame technique enables architectures of visually nested HTML pages. It is often used for page layout purposes and to optimize the speed of page presentation. Common uses comprise the maintenance of a table of content with various structural links in one frame and the presentation of the modifiable content in another frame. When selecting content in the TOC only the content frame is being reloaded, not the TOC itself. The frame technique creates the impression of a visually coherent, monolithic “page”, which is technically made up of several HTML pages. Usually, a frameset is build by one encapsulating HTML page that declares the frameset and two to four (notionally unlimited) contributing sub-frame-pages.

The problem with frame pages is that every participating HTML page will create its own Scone events as soon as it is loaded, resulting in three or more representations in the visualization instead of one. This behavior would undermine the visual metaphor and create unreadable, messed views. It was desirable to create only one visual representation for all pages comprised by a single frameset.

Prototype A addresses this problem with a practical workaround, but does not solve the problem completely. Regarding its behavior, it recognizes framesets and represents them as only one visual node. As soon as a subframepage is replaced, e.g. since the user activated a link in another subframe, this action is treated like a visit to a completely new web page and a new visual representation is inserted in the session graph – referring to the newly modified subframepage. This workaround allows to revisit single subframe pages, but not to reconstruct the complete frameset as originally visited.

Technically every visited page is analyzed immediately after a Scone type-1-event. In case it is a subframe page, detected by a filled `parent_nodeID` field, the URL of its *parent* is added to a ring buffer of size 50. If a subframe page’s parent does not exist in the ringbuffer, a corresponding **WebObject** is immediately created and visualized as regular node. Scone type-1-events that appear within three seconds after the last type-1-event are ignored if their parent is already stored in the ring buffer. This guarantees that if several subframe pages are loaded at once by a frameset, only the first one is considered as an object for visualization.

The implementation is based on the classes **FrameCapsule** and **FrameTimer**. A **FrameCapsule** object is used to store all subframepages that belong to one parent, identified by the parent’s URL (Figure 3-29, line 6-9 and 22).

A **WebObject** is created only for the first encountered subframe per *parent* (line 4). A **FrameTimer** object is instantiated per parent that ensures that all subsequent visits to subframes of this parent are ignored if they occur within a time span of three seconds (Figure 3-29, line 11-16, and Figure 3-30). This time span was chosen as approximation, assuming that all subframes of one parent are loaded within

```

1 // this is the first subframe page to a certain parent
2 // no frame capsule exists
3 if (fcexistent == null) {
4   wo = new WebObject(e.getAccess().getReferrerNode());
5   ...
6   FrameCapsule fc = new FrameCapsule(parent, wo);
7   // add the URL of this subframe to the capsule
8   fc.put(e.getAccess().getFrameName());
9   this.frameCapsules_.put(parent, fc);
10
11  // create timer for this parent (first delete old if existing)
12  if (this.frameTimers_.get(parent) != null) {
13    this.frameTimers_.remove(parent);
14  }
15  FrameTimer ft = new FrameTimer(parent, this.frameCapsules_);
16  this.frameTimers_.put(parent, ft);
17  ...
18 }
19 // if capsule exists - if this is not the first subframe to a parent
20 else if (fcexistent != null) {
21   // add the URL of this subframe to the capsule
22   fcexistent.put(e.getAccess().getFrameName());
23
24 // start over the timer
25 FrameTimer ft = (FrameTimer) this.frameTimers_.get(parent);
26 ft.startOver();
27 }

```

Figure 3-29: Code that treats framesets (1) - SGASconePlugin.update().

three seconds. If the same parent occurs in an event after three seconds or more it is assumed that the user clicked a link in the frameset that caused at least one participating HTML page to change and that the newly visited framepage is rightfully visualized as a new node.¹ Each subframe that appears within the chosen timespan resets the timer again. After expiration the timer removes the related **FrameCapsule** so that a new visit to a subframe of this parent results in the creation of a new **WebObject**, a new visual node, **FrameCapsule** and related **Timer**.

This treatment of frame pages proved to be useful for practical application during the longitudinal study and thereafter. However, it still shows shortcomings as it does not support the detailed reconstruction of a complete frameset. Instead of storing the complete nesting structures just one contributing subframe page is referenced. Only this page will be loaded to the browser when the user initiates a revisitation. It is good enough to prevent history visualizations from being cluttered by numerous visual objects representing all contributing subframepages. This work-around can be applied by any history visualization-based on the visualization of single path elements.

¹ This treatment of frame pages was also used in our web usage study (chapter 5) in order to treat a visited frameset as one coherent entity. The chosen value of three seconds was applied here for the German participants and showed feasible results. Since the Dutch users accessed the web via a faster line we could reduce this value to 1.8 second for them.

```

1 public class FrameTimer {
2     private String _startDate;
3     int milliseconds = 3000;
4     private MeasureTask _measureTask=null;
5     private java.util.Timer _timer;
6     protected static String _parentFrameName="";
7     protected static Hashtable _frameCapsules;
8
9     public FrameTimer(String parentFrameName, Hashtable frameCapsules) {
10        this._parentFrameName = parentFrameName;
11        this._frameCapsules = frameCapsules;
12        this._startDate = DateUtils.getDate();
13        startOver();
14    }
15    public void startOver() {
16        if (_measureTask != null) {
17            _measureTask.cancel();
18        }
19        if (_timer == null) {
20            _timer = new java.util.Timer();
21        }
22        _measureTask = new MeasureTask();
23        _timer.schedule(_measureTask, milliseconds);
24    }
25    class MeasureTask extends TimerTask {
26        public void run() {
27            _frameCapsules.remove(_parentFrameName);
28            _timer.cancel();
29            _timer=null;
30        }
31    }
32 }

```

Figure 3-30: Code that treats framesets (2) - The timer .

Causes the system to wait for subsequent frame pages that belong to the same frameset.

Persistence Concepts

For five rather different purposes persistence mechanisms were required – compare right side of Figure 3-15:

- 1) *History visualizations* created by the SessionGraphs prototype –including information about visited web pages, the path, modified and color-coded visualizations, zoom and pan states of single views.
- 2) *User preference files* used to determine the UI and behavior of SessionGraphs.
- 3) *Thumbnails* of visited web pages.
- 4) *SG-session information for exchange* amongst different users.

Where applicable, a straight forward, file-based method was chosen that fully served the needs of this prototypical solution.

Persistent History Visualizations

The `sga.io.HistoryPersistenceManager` coordinates history persistency. It uses a reader (`SGAFileReaderMultiFiles`) and a writer (`SGAFileWriterMultiFiles`). All classes that model visual entities, such as `VisNode`, `VisEdge`, `SessionGraph`, etc. implement a `toFileString` method that is responsible to create the respective part of a text-based representation in order to write all visual attributes to a text file whenever the application is closed, but also whenever the current SG-session changes or when the user manually saves the current state.

An XML-based format has been chosen for persistency, with some modifications to the syntax. Figure 3-31 shows clippings of an SG-session's file representation. In lines 3 to 8 values are stored that refer to an SG-session including the last camera position before saving. Subsequently, the session graph and the visits list are represented. Line 12 to 35 represent a `VisNode` including the web page and visualization information, line 37 to 41 represent a `VisEdge`. Each visit is stored as a combination of web page and visit specific data (line 45 to 63). To be able to reconstruct the relation between `VisNodes` and `VisEdges` their java object ID is stored as identifier (line 4, 38, 39).

The same history persistence mechanism is also used to provide the prototype's undo-redo functionality. It enables the user to undo and redo modifications of the history visualizations such as the addition, deletion, or the repositioning of nodes. For this purpose, twenty complete history states are persistently organized in a circular queue realized by twenty subdirectories on the local hard disk. Global counters indicate the up to date subdirectory as well as the first and last valid directory. When the user selects the undo function, the counter is decreased by one and the resulting directory is read as the new state. When all directories contain valid histories this can be done twenty times sequentially. The redo function allows stepping forward in this circular queue of states.

An example of the local file organization is shown in Figure 3-32. All files are located in a directory "user_history_bi" under the installation directory. Each history state is represented as a system of files and directories for SG-sessions and SG-tasks. The example in Figure 3-32 shows a directory that contains all information about an SG-task "seminare". The file name is enriched with an ID necessary for internal processing. The directory contains four files that specify four SG-sessions including "Google 6", "google_ikea.de", and "Outdoor". The ".order" file stores the sequence of the SG-sessions in accordance with the enclosing SG-task. This enables the file reader to reconstruct the intended sequence during parsing rather than an alphabetical one. The ".taskView" file holds the graphical layout information for the related task view (miniature map layout).


```

1 This file was generated by SGAFFileWriterMultiFiles Version biv_ioMultiFiles 0.16
2 <SessionNode>
3   title:           "google_hcil"
4   s_tempid:        "sga.sessions.SessionRepresentation@71949b"
5   is current session: "true"
6   id:              ".12"
7   SessionTitleGivenByUser: "false"
8   CameraAffineTransform:"AffineTransform[[1.0, 0.0, 0.0], [0.0, 1.0, 77.2973]]"
9   <SessionGraph>
10    temp_id:         "sga.sessions.sessiongraphs.SessionGraph@52eef4"
11    currentNode:     "http://www.cs.umd.edu/hcil/"
12    <VisNode>
13      s_title:       "Maryland, H."
14      temp_id:       "sga.sessions.sessiongraphs.VisNode@c74910"
15      numVis:        "1"
16      acStayTime:    "0"
17      creationTime:  "1122065392373"
18      userCol:       "null"
19      x:              "20.848800603374894"
20      y:              "56.78396949377771"
21      <WebObject>
22        title:       "University of Maryland, Human-Computer Interaction Lab"
23        origURL:     "http://www.cs.umd.edu/hcil/"
24        numOfOutLinks: "98"
25        numOfOutExternalLinks: "18"
26        isFramePage: "false"
27        hashDomain:  "F75119F6A874AAB9"
28        hashURL:     "079AB94B82B2F8B7"
29        frameChildrenNames: ""
30        numOfWorks:  "520"
31        numOfParagraphs: "21"
32        thumbFileString: "hcil.jpeg"
33        thumbDirString_: "www.cs.umd.edu"
34      </WebObject>
35    </VisNode>
36    ...
37    <VisEdge>
38      from:          "sga.sessions.sessiongraphs.VisNode@c74910"
39      to:            "sga.sessions.sessiongraphs.VisNode@18dfa76"
40      direction:    "4"
41    </VisEdge>
42    ...
43  </SessionGraph>
44  <SessionEntries>
45    <Visit>
46      <WebObject>
47        title:       "University of Maryland, Human-Computer Interaction Lab"
48        origURL:     "http://www.cs.umd.edu/hcil/"
49        numOfOutLinks: "98"
50        numOfOutExternalLinks: "18"
51        isFramePage: "false"
52        hashDomain:  "F75119F6A874AAB9"
53        hashURL:     "079AB94B82B2F8B7"
54        frameChildrenNames: ""
55        numOfWorks:  "520"
56        numOfParagraphs: "21"
57        thumbFileString: "hcil.jpeg"
58        thumbDirString_: "www.cs.umd.edu"
59      </WebObject>
60      user:          "null"
61      timeEntered:  "0"
62      frameId:      "SCONE11220653754088778"
63    </Visit>
64  </SessionEntries>
65 </SessionNode>

```

Figure 3-31: Excerpt of a file that stores SG-session visualization data.

Persistent User Settings

Persistency was also necessary for storing user defined preferences. All settings are stored in files, parsed on system startup and written back once changed. The class

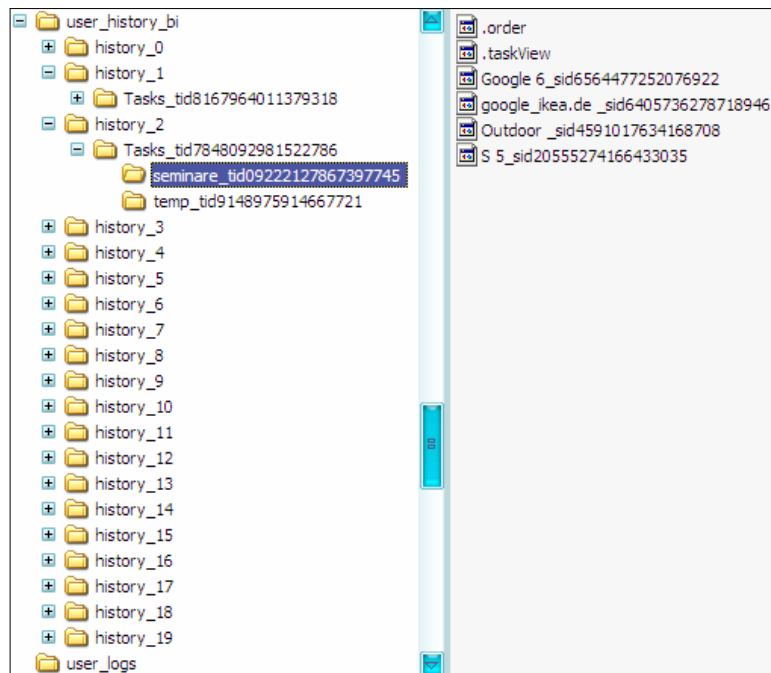


Figure 3-32: File structure used to store the history information persistently.

`sga.config.ConfigMain` is responsible for managing these settings. Using the settings configuration dialog (see Figure 3-11, top left) users can create and switch between various sets of individual settings. Each set of settings is stored in a separate textual file (see Figure 3-33 for an example).

For this purpose, specific classes were created that model *color* and *boolean* values as well as *int* and *string* values. Objects of these classes can be modified during life time via setter methods, they can be cloned for duplication of settings, and they support output methods to create persistent string representations (**DoubleObject**, **IntObject**, **ColorObject**, etc.).

Persistency for Group Exchange

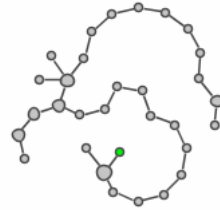
Finally, the exchange of session related information within a workgroup required persistency. The current implementation allows sending and receiving session visualization by email so that the receiver is able to use it as if it was his or her own session. The implementation reuses code of **HistoryPersistenceManager**. Only an additional method **writeSessionForEmail** was added. When the user selects the “mail session to a friend” from the menu, the current SG-session is saved in a dedicated directory (“`user_emailing_sessions/sending`”). To send the data, the user can switch to an email application and attach the generated file to an email and send it to the desired person. The recipient can save the attached SG-session information in the local directory “`user_emailing_sessions/receiving`”. Then he can select “check for new mailed sessions” in the “extras” menu to parse the new SG-session

```
1 This file was generated by ConfigFileWriter Version config 0.1
2 <CurrentConfig>
3   <BACKGROUND_COLOR>           "java.awt.Color[r=246,g=245,b=245]"
4
5   <SESSION_BACKGROUND_COLOR>    "java.awt.Color[r=236,g=235,b=235]"
6   <NODE_BORDER_COLOR>          "java.awt.Color[r=73,g=73,b=73]"
7   <NODE_FILL_COLOR_DEFAULT>     "java.awt.Color[r=197,g=197,b=197]"
8   <NODE_SELECTION_COLOR>       "java.awt.Color[r=0,g=255,b=204]"
9   <NODE_FILL_COLOR_FRONTMOST_WINDOW> "java.awt.Color[r=0,g=245,b=11]"
10 ...
11  <NODE_SIZE>                   "10.0"
12  <MIN_NODE_SIZE>               "8.0"
13  <MAX_NODE_SIZE>               "22.0"
14 ...
15  <LINK_PEN_COLOR>              "java.awt.Color[r=73,g=73,b=73]"
16  <LINK_PEN_WIDTH>              "2.0"
17  <DRAW DIRECTIONS>             "false"
18  <ANIMATE_VIEWS>               "true"
19 ...
20 </CurrentConfig>
```

Figure 3-33: User definable GUI settings stored in a file.

information. It will be added to the local SG-task “temp”. The received SG-session replaces the currently active SG-session and is visualized instantly. Using this proceeding, individuals can exchange visual SG-session information in a simple, basic way.

This chapter presented the straightforward conceptual and technical design and implementation of the SessionGraphs approach. The creation of the prototypes showed that the approach is technically manageable with currently available technology in a reasonable setup. The following chapter will address the empirical evaluation of the SessionGraphs approach.



4

Empirical Evaluation of the SessionGraphs Approach

*If there were only one truth,
you couldn't paint a hundred canvases on the same theme.
Pablo Picasso*

This chapter presents the empirical evaluation of specific aspects of the SessionGraphs approach based on two controlled experiments and one longitudinal study conducted using two versions of the SessionGraphs prototype.

4.1 Methodical Approach

Three main methodical approaches can be considered for studying web navigation and revisitation behavior (see also Cockburn & McKenzie 2001).

Query-based techniques such as *questionnaires* and *interviews*, used for example in the GVU, UCLA and UCS surveys (cf. chapter 2.1.1), are a valuable means for gathering subjective statements from a large number of users. They can be used not only to gain insights about background demographics, but also about detailed aspects of the users' conscious interaction with the web. The drawback is that only information about conscious, perceived behavior will be retrieved, and not about the actual inter-

action. Furthermore, no information is collected about the actual interaction with the tool that could be used to derive navigation patterns and revisitation behavior.

Second, *accompanying observation techniques* (controlled experiments, think-aloud studies, and ethnographic methods) can be used to observe the user during the execution of his or her activities. Controlled experiments are primarily used to examine 'micro' issues, e.g. the models that users apply when using the back button (Cockburn & Jones 1996). The quality of results is strongly determined by the adequacy of the pre-defined tasks – thus, they have to be carefully designed to represent important aspects of the user's daily work. Think aloud studies broaden the scope to 'macro' issues. An example is the analysis by Byrne et al. (1999) who studied people and their tasks while they were using the web. Ethnographic methods, used for example by (Bellotti & Rogers 1997), broaden the scope even further to include also such activities that are not primarily web-based. These are the most time-consuming and long-term methods among the three accompanying observation techniques.

The third technique is *retrospective observation of user actions* – in the context of the web, these are *client-side log-file analysis* methods. They can reveal insights about the actual usage of browser features and (re-)visitation patterns. They allow observation under much more naturalistic conditions and provide insights about coherent, detailed web usage patterns, which is not possible by applying the more common server-side logging techniques (Jansen, Spink & Saracevic 2000; Spiliopoulou, Mobasher, Berendt & Nakagawa 2003; Pitkow 1998).¹

To evaluate the SessionGraphs concept, an adequate combination of the previously mentioned techniques was chosen. Both the benefits of accompanying and retrospective observation were desired, therefore, two controlled experiments and a longitudinal study have been performed. The controlled experiments were designed to reveal measurable, both quantitative and qualitative, insights about the usage of session visualizations for short and long-term revisitations under controlled conditions using predefined tasks. The retrospective longitudinal case study was designed to reveal qualitative insights about the usage of the SessionGraphs prototype under more naturalistic circumstances, including the usage of SG-sessions and SG-tasks in day-to-day activities. Both for the controlled experiments and the longitudinal study, *client-side clickstream logs* have been used to protocol the detailed sequence of the participants' user interface activities, as well as *interviews* to reveal more information about the subjective motivation for certain behavior. The questionnaires used for the interviews can be found in appendix 9.5.

¹ These techniques are generally used to inform site-designers about the usage of their site, but have also been used to create server specific navigation aids, such as Footprints (compare p. 58).

4.2 Two Controlled Experiments

The same group of ten participants joined two controlled experiments, which were designed to compare SessionGraphs' session visualizations with a common browser when used for short and long-term revisitations. The first experiment focused on the examination of *short-term* revisits using a predecessor of SessionGraphs prototype A, and compared it to revisits using a plain Netscape Navigator 4.7. In the second experiment, subjects were asked to revisit the same pages visited one to six days earlier in the first experiment in order to examine the support of *long-term* revisits. Both experiments did not address the evaluation of the task concept, since this would have required a more naturalistic scenario that lies beyond the scope of such time-constrained, controlled experiments.

The prototype used for these two experiments differed from the described prototype A (chapter 3.3) in the following: (1) no "liquidity factor" was used, the session visualizations were always "liquid"; (2) no "suspended" mode allowed suppressing tracking; and, (3) node titles consisted simply of the first three words of a page's title. The later modifications within these three areas were motivated by the findings made in the two experiments and in additional informal feedback sessions.

Both experiments were conducted in April 2001 during a three months research visit at the HCIL, University of Maryland, in close cooperation with Ben Bederson (see Mayer & Bederson 2001).

The hypotheses were that users who use SessionGraphs would be *faster* and *visit less intermediary navigation pages* when revisiting a certain page, and also that they would be *more satisfied* using SessionGraphs than users who only use the history tools of a plain Netscape Navigator.

4.2.1 Participants, Equipment, and Stimuli

All ten subjects were computer science students at the University of Maryland, and all ten participated in both experiments. Six of them were between 20 and 29 years old, three between 30 and 39, and one between 40 and 49. All participants had used Netscape Navigator before. Five subjects reported they used both Internet Explorer and Netscape Navigator as their regular browser, two only Netscape Navigator, and three only Internet Explorer. Four subjects reported using the web between 10 and 19 hours per week; six said they use it for more than 20 hours. Nine subjects majored in computer science, one in information science. Four subjects were female, six male. All subjects were offered payment for their participation.

The equipment was conducted on a 600 MHz Windows NT PC with 512MB RAM, a 21-inch monitor with 1280 x 1024 pixel resolution, and a leased line internet connection. A small Java application presented the questions (see Figure 4-1 for

Screen 15/38

Task 5

Now we begin to revisit some pages you already have visited.

Go back to the "Travel Basics" at Colorado National Monument.

The "Weather and Climate" section says how warm the summers are. How warm are they?

Start Browsing - Start Timer

60 degrees to 110 degrees

75 degrees to 100 degrees

85 degrees to 100 degrees

90 degrees to 110 degrees

80 degrees to 100 degrees

Press the 'submit' button when you have finished your comments.

Submit your answer - Stop Timer

Figure 4-1: Exemplary study task as presented on screen.

When the subject clicked the start button, a timer started and the application switched to the browser. Once the answer was found, the subject entered it and thereby stopped the timer.

an example) and recorded statistics in the background. The system measured the *number of page visits* and the *time needed for completing a specific task*. During task execution, users also had access to a paper version of each task so they did not have to switch applications if they forgot a certain part of the task. One major reason for choosing the Netscape Navigator 4.7 browser for comparison was that it allowed the browser's internal history (of every user) to be stored separately as a file, and then be reloaded into the browser before the second meeting.

The stimuli chosen were web pages of the U.S. National Park Service website (www.nps.gov, see Figure 4-2 to Figure 4-4). This site was selected particularly since it contained several similar web pages of comparable size and content structure that were well suited to formulate related experimental tasks. The site contained pages for all national parks in the United States, structured in the common hierarchical manner. They were comprised of text, images and different sets of links.

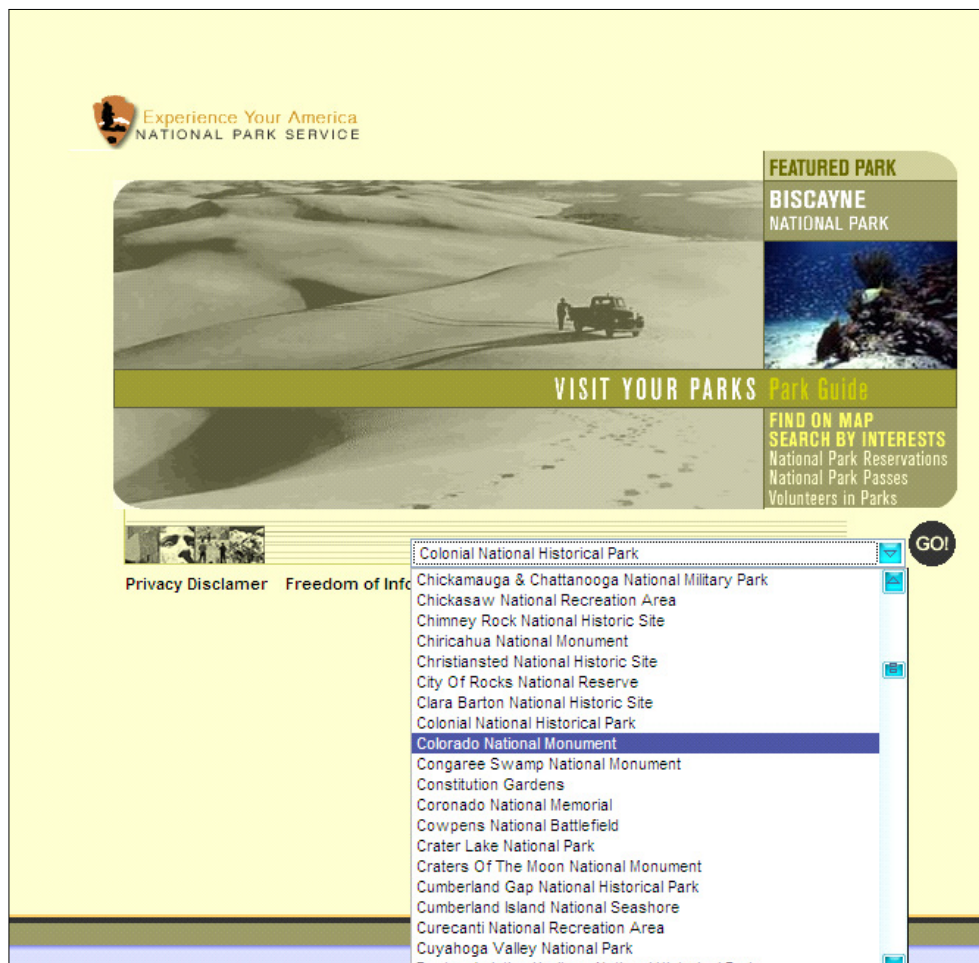


Figure 4-2: Home page of the National Park Service as used for the study. Specific park pages were accessible via a pulldown menu.

4.2.2 Controlled Experiment 1 - Short-Term Revisits

The first controlled experiment aimed at examining the support of *short-term revisitations* offered by the session and task views.

4.2.2.1 Method and Procedure

All participants had to complete two similar sets of tasks, labeled A and B. For one set, they would use Netscape Navigator alone. For the other, they would use the SessionGraphs tool (then, still named “Browsing Icons”) next to the Netscape Navigator window.¹ Each set started with four pretasks in which the users had to

¹ The tool then was still labelled “Browsing Icons”. According to the feedback of several users, this title was misleading and changed to SessionGraphs.



Figure 4-3: Example of specific park page.

The “Travel Basics” link (upper left corner) is required to solve the task presented in Figure 4-1.

visit four pages for the first time, each two levels deep in the site’s hierarchy. Subsequently, they had to revisit each of these pages in four experimental tasks using the history mechanisms of either Netscape Navigator or SessionGraphs. A fifth question requested comparing three pages previously visited, located one level below the NPS start page. Example tasks included “Go back to the ‘Fort Washington’ ‘Activities’ page. What phone number is listed at the bottom?” or “Compare the NPS home pages for the following three parks. Each of them shows a photo of the park in the upper right corner. How many of the photos show people on them?”. The task phrasings precisely described what to look for and how to get there so that the variability due to differences in finding a certain link or target were small. Once the solution was found, the participants would switch back to the evaluation program and stop the time counter by entering the solution. The first meeting took approximately one hour for each subject including introduction, training, execution of tasks and answering the questionnaire.

The screenshot shows a webpage for the Colorado National Monument. At the top, it features the National Park Service logo and the text 'experience YOUR AMERICA'. The main heading is 'Colorado National Monument'. Below this, there is a 'TRAVEL BASICS' section with a navigation menu: 'HOME | CAMPING | ACTIVITIES | FACILITIES | FEES/PERMITS'. The page is divided into several sections:

- Operating Hours, Seasons:** Park is open 24 hours a day. The Visitor Center is open every day, except Christmas. Visitor Center winter hours: 9:00 a.m. to 5:00 p.m.
- Getting There:**
 - PLANE:** Grand Junction Airport: Sky West, United Express, and America West.
 - CAR:** East to West: Highway I-70 to Grand Junction, exit at Horizon Drive, Exit 31 to East Entrance. West to East: Highway I-70 to Fruita, Colorado, Exit 19 to West Entrance.
 - PUBLIC TRANSPORTATION:** Tour bus service available. Contact Grand Junction Visitor and Convention bureau for more information. Amtrack provides services to Grand Junction every day.
- Weather & Climate:** Summers are hot and dry, 80 degrees to 100 degrees. Winters are mild, 20 degrees to 45 degrees.
- Accessibility:** The Visitor Center is accessible, as well as one campsite in the campground, three overlooks, and Devils Kitchen picnic area on the East side of the Monument.
- Getting Around:** The main road is historic Rim Rock Drive, consisting of 23 miles of paved road. Also available is one nature trail and 13 backcountry trails (various lengths), elevation 4,700' to 7,028'.

On the right side of the page, there is a vertical menu with links: 'MAPS', 'BOOKSTORE', 'CONTACT LINKS', 'VOLUNTEER FACTS/DOCS', and a 'for KIDS' button. At the bottom, there are links for 'Privacy Disclaimer', 'Freedom of Information Act', 'park guide', 'search', and 'main'.

Figure 4-4: Typical target page containing the solution to one task. This example shows the answer to the task presented in Figure 4-1.

A 2x2 block design was used for this experiment which enabled collecting more information from each subject than a between-subjects design would have allowed. It also reduced some of the between-subjects variability. The independent variables were the browsing technique that was used first (SessionGraphs or Netscape Navigator alone) and the set of tasks that was used first (A or B). Each subject was randomly assigned to one of four groups:

- Group 1: first tasks A + SessionGraphs, then tasks B + Netscape.
- Group 2: first tasks A + Netscape, then tasks B + SessionGraphs.
- Group 3: first tasks B + SessionGraphs, then tasks A + Netscape.
- Group 4: first tasks B + Netscape, then tasks A + SessionGraphs.

The dependent variables were the average time for task completion, the average number of pages visited until task completion, as well as several user satisfaction ratings. User satisfaction was measured using specific questions in the style of QUIS questions (Questionnaire for User Interaction Satisfaction), developed at the

University of Maryland (Slaughter, Harper & Norman 1994). Its scale ranges from 1 (negative extreme) to 9 (positive extreme).

Time spent on pretasks was not considered since these activities did not include revisitation. Additionally, all subjects answered all questions correctly so all results could be counted.

Training

Each subject was introduced to both systems before executing the tasks. Introduction to Netscape Navigator included a demonstration of the Back and Forward Buttons, the manipulation of bookmarks, and the use of the history list, which several participants were not familiar with. Introduction to SessionGraphs included a short demo session so that the visual behavior of the session visualizations could be seen. A new SG-session was generated and the interface was explained, followed by a brief demonstration of the text search function. Subsequently, each subject executed four non-evaluated training tasks to get a feeling for the unknown system. They learned how to manipulate session visualizations, create a new SG-session, and load a page in the browser. All subjects were experienced with Netscape Navigator so the decision was made not to have them practice this tool.

The NPS home page provides a dropdown menu to select the park of interest. All subjects were introduced to the different possibilities for navigation within this menu, for example, utilizing key shortcuts.

For the Netscape tasks, subjects were encouraged to use any history functionality they considered appropriate, including bookmarks and the history list. To keep settings comparable, subjects were asked to use just a single browser window.

Each participant's Netscape history was empty in the beginning and the bookmarks contained only the default bookmarks from browser setup. The personal toolbar was enriched with one bookmark that directly led to the NPS start page. Subjects were informed that they could use this toolbar button to return to the page whenever they wanted.

4.2.2.2 Results for Short-Term Revisits

Paired t-tests were executed on the averages in order to calculate the results (see Table 4-1 and Figure 4-5). For all the comparisons related to the short and long-term revisits, p was much lower than $\alpha = 0.05$, so all related results can be considered statistically significant. The results, however, showed high variances and the number of participants was relatively small. Variances in task completion times were amplified by changing delays on the network. Consequently, the results should be interpreted with care.

Subjects showed a significantly faster task completion time for short-term revisits in this experiment; they were able to accomplish short-term revisits in 84% of the time with the SessionGraphs tool than they needed when using Netscape Navigator alone.

The user satisfaction questions showed that the participants clearly preferred SessionGraphs over Netscape Navigator. To quantify user satisfaction, several questions were asked per topic and later compiled into a single mean value. To receive two values that represent a subject's overall impression of both tools, for example, he or she was asked to rank both systems along six dimensions, each with a potential value from one to nine. The dimensions in this example were "terrible .. wonderful", "frustrating .. satisfying", "useless .. useful", "difficult to use .. easy to use", "difficult to learn .. easy to learn", and "inadequate power .. adequate power". The final average was derived as the mean of each subject's average rating (see appendix 9.4 for the detailed questionnaire). While the subjects rated the Netscape tool on average with a value of 5.8 (negative: 1 – positive: 9), the SessionGraphs tool received a mean value of 7.6 for short-term revisitation support. In addition, the estimations of the history support features of SessionGraphs and their visual design were significantly higher (7.9 compared to 5.6 & 7.5 compared to 5.6, cf. Table 4-1).

4.2.3 Controlled Experiment 2 - Long-Term Revisits

To obtain detailed information on the support of long-term revisits, a second experiment was conducted one to six days after the first meeting, consonant to the subjects' schedules.

Due to a major difference in the behavior of SessionGraphs and Netscape Navigator, different results compared to the first experiment were expected: Netscape Navigator emptied its Go-menu, as well as the Back and Forward Buttons as it does when the application is closed. In contrast, SessionGraphs provided exactly the same history information as before. Different results were also expected for SessionGraphs considering that some time had passed since the graphs had been generated and last viewed.

After experiment one, each user's Netscape profile was saved along with the SessionGraphs history. Prior to the second meeting, the Netscape profile and SessionGraphs history data were reloaded for each user. To provide more realistic circumstances, approximately 150 entries were added to each participant's Netscape history list, representing pages that had meanwhile been visited. The average number of new pages visited per day, 42, was estimated following (Cockburn & McKenzie 2001). No entries were added to the SessionGraphs history since they would have been stored in different tasks within the hierarchy, and therefore play no role when work resumes on the required tasks.

	SessionGraphs		Netscape	
	Mean	SD	Mean	SD
Required Time				
For Short-Term Revisits (seconds)	35.5	12.2	42.3	19.6
For Long-Term Revisits (seconds)	36.7	15.8	63.5	28.6
Required Number of Intermediary Pages				
During Long-Term Revisits (num. of pages)	1.9	1.1	3.5	2.1
Subjective Ratings (QUIS) Values 1 (negative extreme) to 9 (positive extreme)				
Subjective estimation of each system (general)	7.6	1.1	5.8	1.5
Estimation of each tool's revisitation support - after meeting one (Short-Term Revisitation)	7.9	1.2	5.6	1.8
Visual design of history elements (small & large graphs/ Back, Bookmarks, History List)	7.5	1.2	5.6	1.8
Estimation of revisitation aspects after second meeting (Long-Term Revisitation)	8.1	0.9	4.7	1.8
Mean of all Subjective Ratings	7.8	1.1	5.4	1.8

Table 4-1: Results of controlled experiments 1 & 2 (see Figure 4-5 for diagram). SessionGraphs outranged Netscape Navigator significantly both when executing short-term and long-term revisitations (for all comparisons: $p < 0.05$).

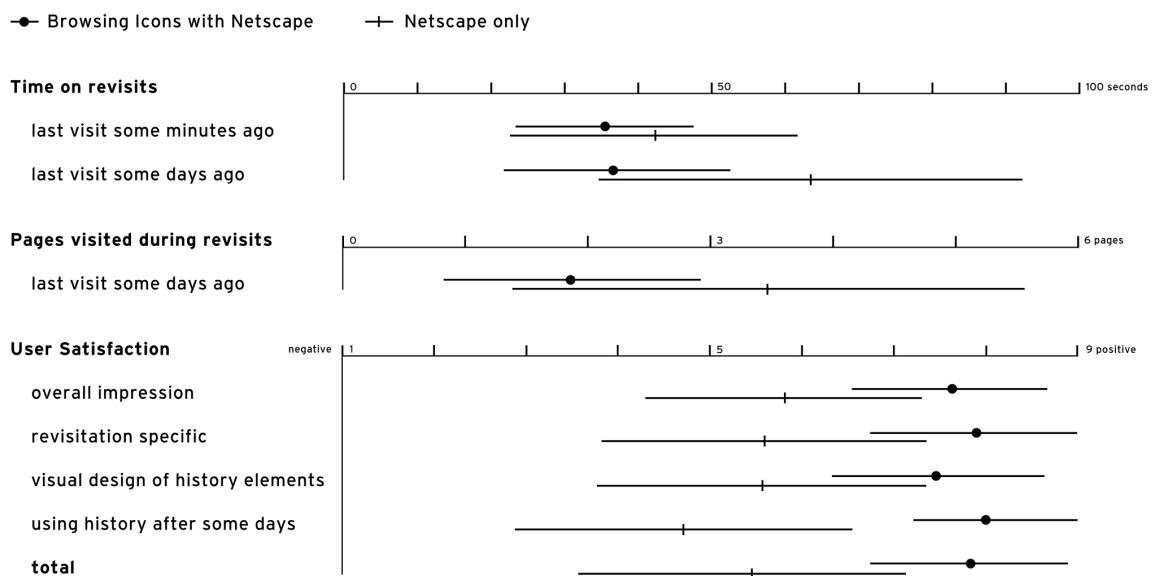


Figure 4-5: Results of controlled experiments 1 & 2 (diagram). For each attribute, this figure shows the mean and the range of (mean - 1 stdev) to (mean + 1 stdev). The tool was still called Browsing Icons then.

4.2.3.1 Method

The same method used in experiment one was applied – each user had to complete three revisit tasks and one compare task that included several revisits per system (SessionGraphs and Netscape Navigator). The execution of the tasks included the revisitation of pages that had already been visited in experiment one. This second meeting took circa 10 to 20 minutes.

A 2x2 block design was also used for this experiment. The dependent variables were the same as in experiment one – average time for task completion, average number of pages visited until completion, as well as a user satisfaction rating that addressed the experiences made with long-term revisits. No training was necessary since all subjects had used both systems before. However, subjects were informed that some pages had been added to their history list in Netscape Navigator in the meantime.

4.2.3.2 Results

The gaps between the measures found for SessionGraphs and Netscape Navigator in this second experiment were even higher than in experiment one. When using SessionGraphs for these long-term revisits, users were almost twice as fast and used only about half the pages during revisits compared to Netscape Navigator. Specifically, for long-term revisits subjects needed only 57.8% of the time, compared to using Netscape Navigator, and they visited only 53.8% of pages during a revisit – p was in each case much lower than 0.01, so also these results are statistically significant.¹ The total time spent for revisitation using SessionGraphs was far below the time generated using Netscape Navigator.

Regarding user satisfaction, the difference between SessionGraphs and Netscape Navigator also increased. All ten users reported that they liked SessionGraphs more (average rating: 8.1) than Netscape Navigator (average rating: 4.7) for long-term revisits (after some days had passed).

A possible explanation for this result is that Netscape Navigator had lost its “short-term memory” – no page could be revisited by using the back button or the go-menu anymore. In contrast, SessionGraphs provided all history entries exactly as laid out before.

We asked the subjects whether they were able to recognize the session visualizations that they had created earlier. Eight of the ten explicitly confirmed this, one of the two remaining participants said that naming the miniature graphs would help to identify them, and the other said he solely used page titles for recognition. These

¹ The number of pages visited could not be used in this experiment due to an error produced by the proxy, which was fixed for the second experiment.

answers and the measured results give evidence that most subjects were able to recognize their session visualizations well, even after several days.

Valuable insights were also gained by subjective statements collected from the participants during both meetings, by answering several guiding (see appendix 9.5). These statements first support the subjective QUIS ratings, since they show that the tool was generally accepted and welcomed by the users. When asked for a general statement about the tool, several users emotionally answered “fun”, or “cool”. Others underlined the practical use of it – “Pretty useful”, “more intuitive than Netscape”, and “much faster and easier”. Furthermore, they pointed to other aspects that were currently not satisfying with the SessionGraphs tool and gave recommendations for further improvement. Several users mentioned that the layout algorithm should be improved in order to reduce crossings of labels and edges, and that node titles should be more meaningful. Controversial feedback was given for the session and task concepts. Some users commended the session view and the ability to visually distinguish between separate activities. Others mentioned that concepts of tasks and sessions – and the difference between both – were not clear enough. In addition, while some were disturbed by the space-consuming approach of SessionGraphs and the animation, others valued the ability to see the history at all times and rated the animation as “more ‘fun’ and easy to understand”. This may indicate that less or limited animation may be an approach that is more accommodating to more users.

When asked to list the “positive things” about the tool, they mentioned its usefulness, the overview regarding the recent path, the session concept to keep separate activities separated, and the modest use of colors.

4.2.4 Modifications According to Participants' Feedback

The following modifications were applied after both experiments concluded in response to participants' feedback. They resulted in the final prototype A that was used for the longitudinal case study. (1) Since some users observed that less animation would be helpful and interaction with the visualizations should be easier, the liquidity factor was introduced as described on page 121. (2) Moreover, the three modes were introduced that enable the user to toggle the visualization between logging, not-logging, and hidden mode (page 99). (3) The node title creation was modified to create shorter, more meaningful titles as described on page 80.

Several minor modifications have been added that were also based on participants' suggestions: The ability to color nodes manually to create more characteristic views; the ability to display the directions of the edges; and, the ability to distinguish hub nodes from other nodes visually (using the “satellites”, page 116).

4.2.5 Limitations of the Experiments

Any interpretation of the quantitative and qualitative results should reflect the constraints and limitations of both experiments – the relatively small sample size and the high variances. Even if all results showed statistical significance, they still have to be interpreted with care. Limitations include the very specific navigation patterns shaped by the designed tasks and the restriction to a single browser window. Such restrictions, however, cannot be avoided in controlled experiments – on the contrary, they are necessary for providing comparable, repeatable results.

An interesting additional analysis would have been the subjective duration assessment, as proposed by Czerwinski et al. (2001, compare page 88). If users would subjectively overestimate task completion time, this would be an indication that the perception and maintenance of the SessionGraphs tool is a distracting interruption, whereas an underestimation would signal that users are able to maintain a state of flow while using SessionGraphs.

4.2.6 Conclusion

In two controlled experiments the session visualizations of the SessionGraphs tool were evaluated using experimental tasks that required the repeated navigation from a starting page to target pages two levels deeper in a site's hierarchy – both in the short and the long-term. Two sets of tasks had to be executed, one using a common web browser, the other using a web browser accompanied by the SessionGraphs tool. In both settings the browser was constrained to one open window at a time.

Both experiments demonstrated quantitatively measurable advantages of the SessionGraphs visualizations over Netscape Navigator's history tools. Especially the results gained in the second experiment, during long-term revisitation, document a big improvement compared to Netscape's history. Both the quantitative measures of time and visited pages and the subjective ratings far exceeded those of Netscape.

The participants also contributed several valuable hints regarding areas of possible improvement. Their statements also showed that certain aspects of the tool, such as animation and visual presence, are controversially received.

It should be taken into consideration that the conducted experiments compared SessionGraphs with a plain standard browser, Netscape Navigator, and not with another visual history tool. The latter was simply not possible within a reasonable timeframe for the scope of this thesis project. The authors of the PadPrints concept also compared their prototype with the plain Netscape browser and revealed an improvement (compare page 50). It can be argued that the SessionGraphs concept additionally overcomes important shortcomings found in the PadPrints approach, i.e. it does not provide a potentially misleading tree view of the path, it scales with growing paths by providing SG-sessions and SG-tasks, and it provides manual view manipulation to create visual representations that correspond to the users' ideas.

Taking these considerations into account, the gained results can still be interpreted as a positive appreciation of the SessionGraphs visualizations, when compared to currently available history mechanisms.

The three primary results found regarding SessionGraphs are:

- 1) The time necessary for short-term revisitations can significantly be reduced compared to a plain web browser. This effect is even stronger for long-term revisits – regular browsers show deficiencies in adequate support.
- 2) The number of navigation revisits – i.e. visits of intermediary navigation pages (compare page 31) – was significantly reduced. While techniques that users applied with the regular browser involved revisits to several intermediary navigation pages, the use of session graphs allowed direct access to a desired page. This is certainly dependent on the task and on the number of overleaped pages. The maximum depth, 2, modeled by the experiments' tasks is still a rather cautiously chosen value. Even higher benefits can be expected under circumstances that involve targets located deeper in a site's hierarchy.
- 3) The subjective satisfaction related to revisitations during information gathering was significantly increased by the SessionGraphs tool. The gap between SessionGraphs and Netscape Navigator was largest for long-term revisits.

While these results were quite promising, they were gained under highly artificial, controlled settings. The next step was to implement the suggested modifications and to evaluate the SessionGraphs tool under more naturalistic conditions over a longer interval of time.

4.3 Longitudinal Study

A longitudinal study should provide insights about how people actually make use of the provided history interface in a naturalistic setting, i.e. working on their individual, day-to-day web activities. A prototype was built that covered the basic requirements of a longitudinal study including most of the functions defined in the concept and the necessary logging abilities to conduct posterior quantitative analyzes on the single events. It was far from being a product ready for market, but included the improvements demanded by participants of the controlled experiments. The study

was not primarily designed to reveal quantitative results, but qualitative ones. The goal was to gain a general impression about the acceptance of the tool under naturalistic circumstances, as well as specific recommendations of the participants for further improvements.

4.3.1 Participants, Prototype, and Equipment

Five unpaid participants, 22 to 37 years of age (mean 29.4), used prototype A in their daily private and / or working context over a period of 74 to 170 days (mean 114.4). Within these periods, they actually used the prototype on 22 to 40 days (mean 32.0). Such days with at least one event of tool usage will be called *active days* in the following. In two interviews, one soon after the beginning and one at the end of the logging period, participants were asked about their impression of the SessionGraphs prototype and concepts. Regarding this certainly very low number of users, Jakob Nielsen (2003) claims “To identify a design’s most important usability problems, testing five users is typically enough”, and this was a major goal of this study.

I also used my own log file of interactions. No personal statements or ratings of me were considered, due to potential bias. Since I used the web for my regular day-to-day work, as the other participants did, there were no reasons not to include these data into the foundation for later evaluation. Furthermore, using the prototype myself not only gave me deeper, more precise knowledge about the tool’s benefits, but also about the current technical and conceptual shortcomings, and it was helpful for better understanding the participants’ statements. The remaining participants were colleagues of mine and students associated to the computer science department in Hamburg. All participants were male except one.

The study took place between December 2002 and August 2003. All participants reported that they were highly experienced with computers and the web. On a QUIS-like scale (compare page 142) they stated the values 7 to 9 for computer experience (mean 8.2) and 8 to 9 for web experience (mean 8.6). They have used computers between 4 and 20 years (mean 11.6 years) and the web for 4 to 10 years (mean 6.4 years). Their weekly time spent on the web – estimated by them – ranged between 5 and 25 hours which comprised both private and professional activities.

Finding participants was somewhat difficult since, in contrast to the controlled experiments, this study required them to change their web-related habits and familiar tools for a considerable amount of time. They additionally were required to switch to Internet Explorer, since an extension was used in order to find out more about the details of browser interaction (three users switched from Netscape Navigator 4.7 or Opera 6.0.5). Furthermore, they had to get used to the SessionGraphs tool and its operation. Additional time was required for the interviews and the installation. They also had to overcome concerns about releasing personal web usage

data. Technical constraints reduced the number of contemplable subjects since the prototype required a Windows platform, admin rights, and a screen size of at least 1024 * 768 pixels. Four interested subjects, for example, were not able to participate in the study either because installation was not possible or the system became too slow. The five remaining participants were able to install and use the prototype without difficulty.

Two participants installed the software on their home computers, two on their office computers, and one on a laptop that was used both at home and at work. Four used it both for professional and private issues, one only for private issues. Operating systems encompassed Windows XP (3), Windows 98 (1), and Windows 2000 (1).

4.3.2 Method and Procedure - Interviews and Logging

All subjects participated voluntarily and were informed to participate at least for five weeks, in case they feel comfortable with the installed software. They were welcome to quit the study without any explanation whenever they wanted. To assist them in installing and using the SessionGraphs prototype, a detailed installation guide and a step-by-step guide to the SessionGraphs interface (appendix 9.5.1) were available online, as well as an online FAQ that included questions arising during the study. Small tutorial screen-capture-videos showed how to use every major aspect and function of the SessionGraphs interface (appendix 9.5.2).

Each of the two interviews lasted for about one hour. The first interview was conducted after a few days of usage. It addressed previous web browsing and revisitation behavior and the first experiences made with the new tool. Users were answered questions they had about tool usage and were introduced into features that they had not explored yet. The second interview was conducted at the end of the logged periods and covered experiences with the tool, subjective estimations, the discussion of specific situations during the study, as well as suggestions for improvements.

The technical setup included the automated maintenance of two log files for retrospective analysis in a directory “\user_logs” under the install-path. The file `longitudinalStudyGlobals.csl` contained general study parameters, such as *user name*, *email*, and *accumulated per day values* such as *total visits per day* or *total daily time spend on the web*. This file gave a quick first impression of a participant’s activities during the study. Figure 4-6 shows an excerpt of such a log file.

Lines 6 to 11 of Figure 4-6 display the duration of accumulated time of prototype usage per mode: `SESSIONGRAPHS-ACTIVE (BiOn_)`, `SESSIONGRAPHS-SUSPENDED (BiOf_)`, and `PLAIN-INTERNET-EXPLORER (Msie)`. A normalized value of accumulated stay-time provided a better approximation of the user’s actual working time, based on phases in which the user actually showed measurable activi-

ties. A related counter paused automatically as soon as the inactive phase after a visit exceeded two minutes. To gain the normalized daily values all the duration was added up in which the counter was running during the day. Unnormalized values represent all the time in which the tool process was running, independent of the user's activity. Line 18, for example, shows that the tool was used on March third for 1:28:27 hours, including 38:03 normalized minutes of actual activity (line 21) and visiting 82 pages (line 36).

The second file `longitudinalStudyProtocolBiv.csl` contained a sequential list of every single visit and every SessionGraphs user interface event (see Figure 4-7).

This detailed log file was used for later analysis of the participants' activities. Line 3 of the exemplary excerpt protocols the system's startup. Next, a session view is opened showing an SG-session "google" (line 4). The visualization was used to revisit a page on `javasite.bme.hu` (line 5). The system registers the new visit when the page is retrieved by the browser (line 6). The user selects two other SG-sessions using the task chooser (lines 7 to 9). He uses the SessionGraphs "find"-function to search the stored history information (line 10). Then he manually moves a node in a view (line 11) and selects it (line 12) to revisit a page on `dict.leo.org` (line 13). He uses the web browser to submit a form and retrieve a search result (line 14). Finally, the system exits.

Preprocessing the Data

The resulting detailed trace of actions stored in this second log file contained different artifacts. In a cleaning process more than 1000 page visits (19.5%) of the initial 5598 visits were identified as artifacts and removed before analysis: several visits were removed that were caused by undetected frame pages, by adservers, by automatically opened windows, and artifacts created by CGI driven websites like eBay. A last modification was that all visits that were caused by the system staring up and loading a predefined home page were tagged and not counted as visits since only actively loaded pages were of interest. The resulting cleaned log files represented more accurately what users actually did. A total of 4546 visits were considered valid visits, in the sense that each single visit was consciously initiated by a user.

```

1 Generated by biv.jar Version 0.9.4 - AutoLogger Version biv_autologger 0.1
2 userEmail_,"mayer@informatik.uni-hamburg.de"
3 userName_,"matthias mayer"
4
5 firstDateOfRunning_,"1046707120236",03/03 04:58:40
6 totalSecondsUnnormalizBiOnUsage_,"93647401", 02:00:47
7 totalSecondsNormalizedBiOnUsage_,"13817971", 03:50:17
8 totalSecondsUnnormalizBiOfUsage_,"319050", 00:05:19
9 totalSecondsNormalizedBiOfUsage_,"239616", 00:03:59
10 totalSecondsUnnormalizMsieUsage_,"0", 00:00:00
11 totalSecondsNormalizedMsieUsage_,"0", 00:00:00
12
13 totalNumberOfVisitsBiOn_,"426"
14 totalNumberOfVisitsBiOf_,"4"
15 totalNumberOfVisitsMsie_,"0"
16 totalNumberOfVisits_,"430"
17
18 dailySecondsUnnormalizBiOnUsage_,"1046707120236", 03/03 04:58:40, *5307442*, 01:28:27
19 dailySecondsUnnormalizBiOnUsage_,"1046774412007", 03/04 11:40:12, *16048548*, 04:27:28
20 ...
21 dailySecondsNormalizedBiOnUsage_,"1046707120236", 03/03 04:58:40, *2283053*, 00:38:03
22 dailySecondsNormalizedBiOnUsage_,"1046774412007", 03/04 11:40:12, *3198230*, 00:53:18
23 ...
24 dailySecondsUnnormalizBiOfUsage_,"1046707120236", 03/03 04:58:40, *0*, 00:00:00
25 dailySecondsUnnormalizBiOfUsage_,"1046774412007", 03/04 11:40:12, *0*, 00:00:00
26 ...
27 dailySecondsNormalizedBiOfUsage_,"1046707120236", 03/03 04:58:40, *0*, 00:00:00
28 dailySecondsNormalizedBiOfUsage_,"1046774412007", 03/04 11:40:12, *0*, 00:00:00
29 ...
30 dailySecondsUnnormalizMsieUsage_,"1046707120236", 03/03 04:58:40, *0*, 00:00:00
31 dailySecondsUnnormalizMsieUsage_,"1046774412007", 03/04 11:40:12, *0*, 00:00:00
32 ...
33 dailySecondsNormalizedMsieUsage_,"1046707120236", 03/03 04:58:40, *0*, 00:00:00
34 dailySecondsNormalizedMsieUsage_,"1046774412007", 03/04 11:40:12, *0*, 00:00:00
35 ...
36 dailyNumberOfVisitsBiOn_,"1046707120236", 03/03 04:58:40, *82*
37 dailyNumberOfVisitsBiOn_,"1046774412007", 03/04 11:40:12, *109*
38 ...
39 dailyNumberOfVisitsBiOf_,"1046707120236", 03/03 04:58:40, *0*
40 dailyNumberOfVisitsBiOf_,"1046774412007", 03/04 11:40:12, *0*
41 ...
42 dailyNumberOfVisitsMsie_,"1046707120236", 03/03 04:58:40, *0*
43 dailyNumberOfVisitsMsie_,"1046774412007", 03/04 11:40:12, *0*
44 ...
45 dailyNumberOfVisits_,"1046707120236", 03/03 04:58:40, *82*
46 dailyNumberOfVisits_,"1046774412007", 03/04 11:40:12, *109*
47 ...
48

```

Figure 4-6: Excerpt of one user's general log file (global and daily values). Colored boxes indicate the parameters and values discussed in the text.

```

1 This file was generated by biv.jar 9.4 - AutoLogger Version biv_autologger 0.1
2 ...
3 1047047406520, 2003-03-07 15:30:06, BU_SYSTEM_STARTING_UP
4 1047047410145, 2003-03-07 15:30:10, B_SESSIONVIEW_OPENED, Google , 0494125010331059
5 1047047423995, 2003-03-07 15:30:23, BU_LOADED_BY_BROWSINGGRAPH, http://javasite.bme.hu/ja
6 1047047424716, 2003-03-07 15:30:24, B_NEW_VISIT_IN_SESSION, http://javasite.bme.hu/ja
7 1047047428792, 2003-03-07 15:30:28, BU_USER_SELECTED_SESSION_BY_TCH, Lycos.de , 10126132150146
8 1047047428792, 2003-03-07 15:30:28, B_SESSIONVIEW_OPENED, Lycos.de , 10126132150146
9 1047049935396, 2003-03-07 16:12:15, B_SESSIONVIEW_OPENED, LEO, 43290060482097303, t
10 1047049936888, 2003-03-07 16:12:16, BU_Find, leo, found: true
11 1047049941865, 2003-03-07 16:12:21, BU_USER_MOVED_SESSIONNODE, from (99.0,30.0), to (163
12 1047049944699, 2003-03-07 16:12:24, BU_LOADED_BY_BROWSINGGRAPH, http://dict.leo.org/?sear
13 1047049947664, 2003-03-07 16:12:27, B_NEW_VISIT_IN_SESSION, http://dict.leo.org/?sear
14 1047049964688, 2003-03-07 16:12:44, MU_LOADED_BY_FORMSUBMIT, http://dict.leo.org/?sear
15 1047052876966, 2003-03-07 17:01:16, BU_SYSTEM_SHUTDOWN
16

```

Figure 4-7: Excerpt of one user's detailed log file (single sequential entries).

4.3.3 Results

A general outcome was that the major functions of the tool were applied successfully by the participants as they were intended by the concept. The study also showed, however, that several technical problems occurred when the prototype was used for actual day-to-day work, and that also the concept has to be further improved in specific areas.

Four of the five participants visited circa 1000 web pages per person, while one created only about 500 page visits (see Table 4-2). The mean number of visits per active day was about 30 for all subjects. As further discussed below, two participants reported that they switched to a regular browser due to technical problems from time to time. The actually measured data are presented next, as they characterize the factual activities of the users during the study period. In the following, subjective comments from the interviews will be mentioned where appropriate and in the section “Insights Based on Interviews & Areas of Improvement” (page 164). The suggestions and comments will be used to create new ideas for improvement, both technically and conceptually.

General Measures	User 1	User 2	User 3	User 4	User 5	<i>All</i>
Total days of participation	170	104	74	142	82	572
Active days	36	22	35	40	27	160
Visits*	935	553	1071	1052	935	4546
Visits before cleaning	1238	673	1200	1295	1192	5598
Visits per active day SESSIONGRAPHS-ACTIVE	26.0	25.1	30.6	26.3	34.6	28.5
Visits per active day SESSIONGRAPHS-SUSPENDED	0.1	2.9	0.4	0.0	0.2	0.7
Visits per active day PLAIN-INTERNET-EXPLORER	0.0	0.0	0.0	0.0	0.0	0.0
Recurrence Rate*	39.9%	34.0%	43.2%	32.8%	46.1%	39.5%
Total Revisits*	371	188	463	345	431	1798

Table 4-2: Results of longitudinal study (general measures).

* These measures refer to the cleaned data, collected in the SessionGraphs-Active mode, free from artifacts like frame pages and adservers.

Interpretation of General Measures

The participants mainly used the tool in the `SESSIONGRAPHS-ACTIVE` mode (compare Table 4-2); it was not used at all in the `PLAIN-INTERNET-EXPLORER` mode. In the interviews, two said they preferred to switch directly to a completely unlogged browser from time to time due to technical problems, but also due to privacy reasons. The `SESSIONGRAPHS-SUSPENDED` mode was only used a few times for short phases in which a non-tracking behavior was preferred (mean over all participants: 0.7 visits per active day). The high usage in the `ACTIVE` mode of four of the five subjects, at least, indicates a good acceptance of the tool. The fifth subject, however, reported that he kept the tool open in the `ACTIVE` mode, but hidden behind the browser and used it only sporadically. When normalized to active days this participant also visited around 30 pages per day using the tool.

Revisitation Behavior

The mean recurrence rate in the cleaned `SESSIONGRAPHS-ACTIVE` data was 39.5% (individual means between 34% and 46%, see Table 4-2). Compared to earlier studies (chapter 2.2.5), this value is relatively low. However, our 2005 study (chapter 5) revealed similar low values for current web use with common browsers. Therefore, this does not necessarily mean that the use of the SessionGraphs prototype leads to less revisits of intermediary navigation pages.

The temporal distribution of revisits followed the expected power law distribution with a large majority of short-term revisits. Three quarters of all revisits occurred within only one hour (see Table 4-3, short-term). About 7% occurred in the medium-term (one hour to one day) and about 16% of all revisits were long-term revisits (after one or more days).

Four out of five revisits (about 80%) were intrasession revisits (see page 31) within the same `SG-session` as the last visit to the same page (see Table 4-4). Interestingly, 7.3% of revisits were intrasession, but medium- or long-term revisits (see page 30), which shows that session visualizations were used successfully as a long-term revisitation and session-resumption tool. The interviews affirmed this intention. Session visualizations were also applied as a tool to keep apart several parallel activities – about 3.5% of all revisits were short-term revisits to another than the current `SG-session`. In this context, one participant mentioned that for this purpose he regularly used the session view tabs to switch between the last three sessions.

Revisitation Details	User 1	User 2	User 3	User 4	User 5	All
Temporal Aspects						
Short-term (% of Rev.)	74.4%	84.6%	67.8%	78.6%	77.3%	76.5%
total	276	159	314	271	333	1353
Medium-term	10.8%	4.3%	10.6%	3.5%	7.4%	7.3%
total	40	8	49	12	32	141
Long-term	14.8%	11.2%	21.6%	18.0%	15.3%	16.2%
total	55	21	100	62	66	304
Aspects of Session Context						
Intrasession (% of Rev.)	83.0%	89.9%	72.8%	78.0%	81.4%	81.0%
total	308	169	337	269	351	1434
Intersession	17.0%	10.1%	27.2%	22.0%	18.6%	19.0%
total	63	19	126	76	80	364

Table 4-3: Revisits - over time and across sessions.

The upper part shows the relation of short-term revisits (within one hour), medium-term revisits (one hour to one day), and long-term revisits (after one day or more). The lower part shows the relation of intra- and intersession revisits according to user defined SG-sessions.

An additional analysis considered the techniques that participants had applied to revisit pages. Of all revisits, 51% were initiated by hyperlinks, a value that is conformable to earlier studies (cf. Table 5-1), and that can be explained by hub and spoke navigations in the short-term and similar-path revisitations in the long-term. Another 10% of all revisits were caused by interaction with the Internet Explorer interface (e.g. by using bookmarks, URL typing, or the history list). The remaining

Revisits: Time & Session	User 1	User 2	User 3	User 4	User 5	All
Short & Intrasession *	72.2%	80.3%	66.7%	76.5%	72.6%	73.7%
total	268	151	309	264	313	1305
Short & Intersession	2.2%	4.3%	1.1%	2.0%	4.6%	2.8%
total	8	8	5	7	20	48
Medium & Intrasession	4.9%	2.7%	5.8%	0.3%	3.3%	3.4%
total	18	5	27	1	14	65
Medium & Intersession	5.9%	1.6%	4.8%	3.2%	4.2%	3.9%
total	22	3	22	11	18	76
Long & Intrasession	5.9%	6.9%	0.2%	1.2%	5.6%	4.0%
total	22	13	1	4	24	64
Long & Intersession	8.9%	4.3%	21.4%	16.8%	9.7%	12.2%
total	33	8	99	58	42	240

Table 4-4: Revisits - interdependency of time and session context.

This table shows that most short-term revisits are also intrasession revisits, and that a considerable amount of intersession revisits are long-term revisits. A substantial amount of intrasession revisits still occurs in the medium and even in the long-term.

* Percentage of all kinds of revisits (cleaned data, SessionGraphs-Active mode).

39% of all revisits were initiated by interacting with the SessionGraphs interface. This high value is a further indication of the good acceptance of the tool.

About four fifths of these SessionGraphs interactions were interactions with session visualizations, about one fifth with task visualizations, the miniature graphs in the lower part of the GUI. This allocation matches the intended design that provides the session view as the major revisitation tool for intra- and parallel-session revisits and the task view for a smaller amount of navigations within related tasks.

These results prove the general acceptance of the SessionGraphs prototype as a history tool capable of satisfying day-to-day needs. The offered strategies for accessing and using the history information were utilized by the participants as intended: session visualizations were intensively used for intra- and parallel-session revisitation, task views were used to switch between related tasks, and also the single nodes in task views were used to directly jump to a node in a related task.

The Use of Sessions and Tasks

In total, the participants created 248 SG-sessions throughout the study (35 to 64 sessions per subject), or a mean of 1.6 sessions per active day and user. The majority of all SG-sessions, about 64%, contained 10 or less nodes (different URIs, see Table 4-5), and half of all sessions used even only 5 nodes or less. It seemed arguable whether visualizations of such small sessions could actually be supportive.

Several participants, however, responded in the interviews that they liked

Nodes in Session	Sessions found (248 in total)	Percentage of Sessions over all subjects
1-10	159	64.1%
11-20	37	14.9%
21-30	25	10.1%
31-40	8	3.2%
41-50	4	1.6%
51-60	6	2.4%
61-70	3	1.2%
71-80	1	0.4%
81-90	1	0.4%
>90	0	0.0%

Table 4-5: Session analysis - number of distinct pages (=nodes) per session.

Distinct Hosts per Session	Sessions found	% of Sessions
1	68	27.4
2-3	73	29.4
4-7	60	24.2
8-21	47	19.0

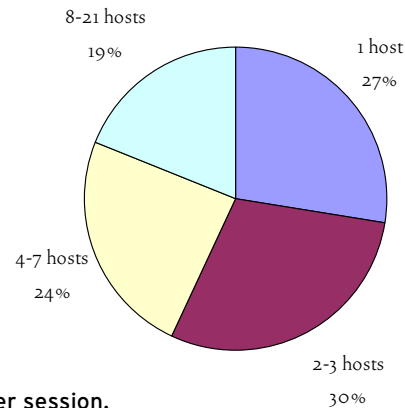


Table 4-6: Session analysis - number of distinct hosts per session.

using the visualizations of small sessions, chiefly for short-term revisits. One also mentioned the advantage over the back button of being able to revisit pages that would otherwise have been popped from the stack. In several cases, small SG-sessions were even used for long-term revisits. On the other hand, some participants mentioned that they believed it made no sense to store so many of the smaller sessions, for instance, in which they had executed simple Google searches. Two explicitly indicated that they preferred using Google than to search for an earlier session visualization in the tasks hierarchy. This leaves room for improving the SessionGraphs approach by offering an easier searchable history, for example.

About a quarter of all SG-sessions contained between 10 and 30 nodes, and still about 10% contained 30 or more nodes. The mean number of nodes in an SG-session was 13.7 – individual means ranged between 10 and 20 nodes. Participants said in the interviews that they appreciated the tool for more demanding, complex sessions, and for sessions with an explicit research character. One said, that he did not like visualizations to be created when he used web-based chat rooms or ICQ channels. This leaves further room for improvement by offering heuristics, for example, that are able to distinguish between different kinds of web usage – that allow reacting in potentially user-definable, different ways.

The participants reported only a small number of session visualizations that contained too many nodes or that became too crowded or complex, except for the participant who used the prototype hidden behind his browser windows – he was complaining about several too large, too complex session visualizations. A more frequently reported problem was that the visualizations did not use screen space adequately, i.e. they started to scroll very early. When participants navigated along a sequence of links and did not manually reshape the visualizations, the layout algorithm would create diagonal, zig-zag shapes (see Figure 4-8a for an example). Two subjects recommended an improvement here, and suggested that the visualizations should grow in a more spiral-like way in order to use space more efficiently and to postpone the point in time when scrolling starts. Exemplary session visualizations are presented in Figure 4-8 and will be discussed below.

A further revealing result was the obvious need for adequate between-site navigation support. Three quarters of all SG-sessions involved visits to more than one host (see Table 4-6). About one third of all SG-sessions involved 2 to 3 different hosts, a quarter 4 to 7 hosts, and a fifth involved even 8 to 21 different hosts. The mean was 5.9 hosts per session. These numbers illustrate the necessity of a history strategy that is able to group activities spread over different websites. None of the currently available tools provide satisfactory support here. Current tools seem to neglect the fact that many sessions involve several hosts. The SessionGraphs approach serves this need by allowing session graphs to contain server-crossing navigation paths.

Exemplary Sessions

The analysis of session visualizations created by the participants during the study gives an impression of how the tool was actually used, and about aspects that still can be improved. Figure 4-8 shows nine exemplary SG-session visualizations. Figure 4-8a shows the exploration of a university department home page and the related search in an online phone book, comprising 12 web pages, or nodes. This session was started from the Google page in the middle. It shows the zig-zag shape resulting from a series of links followed without manipulating the visualization. The upmost phone book page was visited 4 times in total, encoded by the larger size of the node. Figure 4-8b shows a much more manually reshaped visualization of an information gathering process on computational geometry. It comprises 19 web pages, 5 of which were visited more than once. This view was manually reshaped by one of the participants who preferred spiral-like shapes.

Both examples (a and b) show that the automatically derived node titles are still not very meaningful. The modification subsequent to the controlled experiments brought a first improvement, since shorter labels lead to less visual overlaps. They still need to become more expressive. One solution may be to identify constant parts in titles of pages that were consecutively visited on one website, and to omit or abbreviate these constant parts, and to show only the changing parts. Furthermore, the page's content could be considered for deriving a more adequate textual description (see also Xue, Hu, Xin, Song, Shi, Cao, Lin & Li 2007; Hu, Li, Cao, Meyerzon, Teng & Zheng 2006). In the case of searches, the query terms could be used. Session titles could then be derived from the first visited page in a session, and the first session in a task could determine this task's title, still modifiable by the user. Besides displaying a more meaningful title, showing the site's fav-icon or a small thumbnail as a page representation could further increase the readability of the visualizations. The use of thumbnails, however, may have not the intended positive effects, even if often praised by users; Czerwinski et al. (1999) showed that the Data Mountain users found items based on spatial position equally well with and without

thumbnails. Furthermore, when visiting pages of similar content, like pure text on white background, thumbnails fail in their function.

Figure 4-8c, e, and h show typical hub and spoke navigation patterns after a Google search with 16, 18, and 15 nodes. The initially visited Google page forms the center node surrounded by different spokes. The user can immediately go back to a search results list to explore further results, or to the query page to refine a query. Two participants said that the different page types encountered during a search (query formulation, results lists, target pages) should be visually distinguished. One possibility of accomplishing this is presented in chapter 6.1.4.

Figure 4-8d shows a visualization (22 nodes) that grew relatively untouched while later being manually reshaped, mainly in the upper left area. The visualized “404 error” node could be automatically removed based on a heuristic.

Figure 4-8f, g, and i show rather small SG-sessions that comprise 11, 4, and 6 nodes. Figure 4-8f shows a roundtrip through self designed web pages in which the user revisited several nodes repeatedly. The SG-session in Figure 4-8i was used for several revisitations. Figure 4-8g shows a simple Google search, which was one example of the sessions that were considered helpful even despite their small order.

Manipulation and Interaction

Figure 4-9 shows the nineteen most frequent interaction techniques that were used in combination with the SessionGraphs GUI. The most frequently applied interaction was to manually move one or more nodes in a session view, which appeared 561 times in total or about 2.3 times per SG-session and accounted for 17% of all interactions with the SessionGraphs interface (individual means: 9% to 25%). This high share can be interpreted, first, as a sign that participants are willing and able to spend additional efforts to manipulate their visualizations, but also, second, as an indication that the automated visualizations should be improved to relieve the user from having to manipulate them so often. Interestingly, one participant mentioned that for him, it was valuable and pleasant to “play around with the nodes in the view” while a page was loading or while he was planning the next steps of his information gathering process. As already mentioned, however, two users desired a denser auto-layout which postpones the necessity to scroll and gives them more freedom to care about their actual work. It seems worthwhile to follow this latter desire and modify the tool in this sense. Users who love to manipulate the views should still be able to do so.

The second most frequent activity was clicking a node in a session view to revisit the represented web page (532 times in total, mean: 16%, more than twice per SG-session, on average). This is a direct sign of the good acceptance of session visualizations as a history tool, even if very differently applied by the participants (min: 2%, max: 40%).

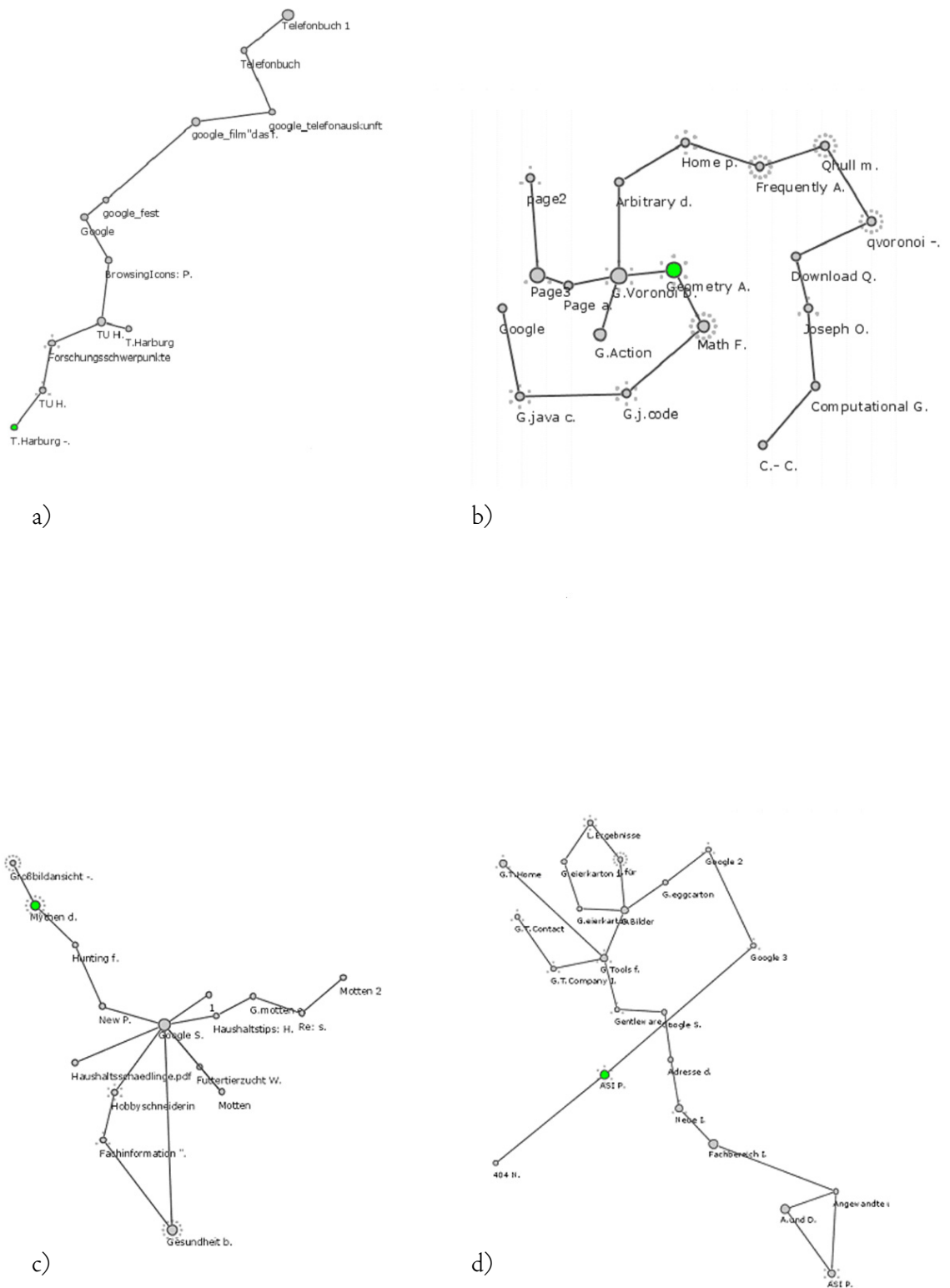
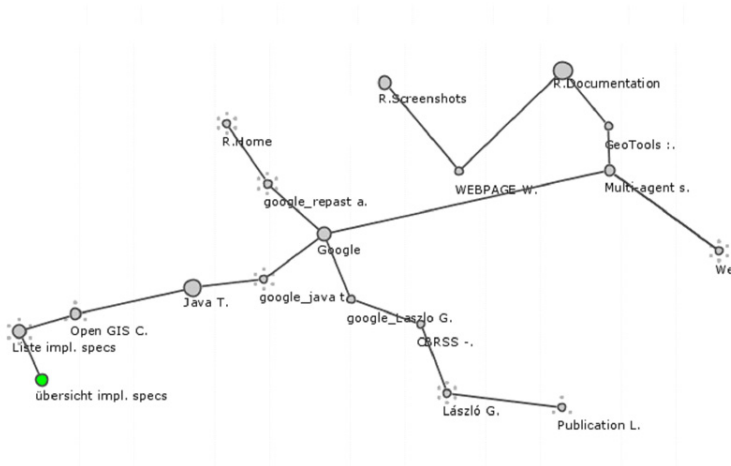


Figure 4-8: Exemplary session visualizations created by the study participants.

a) Exploring a university department b) Search on a computational geometry topic

c) Searching a product to fight a vermin d) Another exploration of a university website



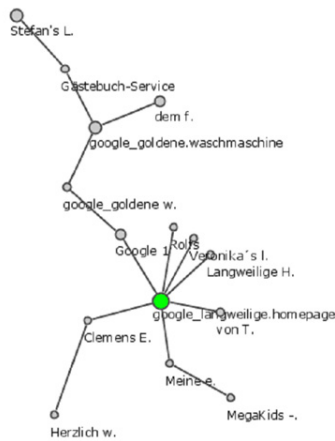
e)



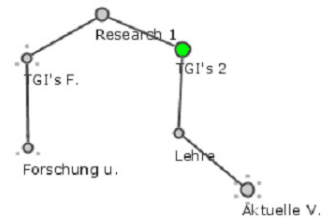
f)



g)



h)



i)

e) Research on multi agent systems f) Exploring self designed pages g) Search for "emo-ticons" h) Exploring home pages (hub & spoke) i) Directed search for a university course

The third most frequent event was the deletion of one or more nodes, indicating that several users wanted to keep their views free from unwanted detail (495 times in total, mean: 15%, min: 0%, max: 40%). One participant mentioned that he would welcome automated help to remove unwanted nodes, i.e. having a process in the background detecting which nodes are really relevant and which are not, in the context of pure navigational pages. Such nodes could then become less visually salient or even disappear completely from the visualization.

Subsequent positions in the frequency list are occupied by different kinds of interaction with the task chooser. This shows that most users accepted the task chooser for session and task organization. One participant, however, stated that he highly preferred using Google to revisit a page in the very long term, than having to navigate the hierarchy of sessions and tasks.

The selection of a session (mean: 13%, min: 6%, max: 29%) was followed by the selection of a task (mean: 6%, min: 2%, max: 15%), which was then followed by interactions with the miniature graphs presented in the task view (moving and clicking a miniature graph – means of 4% each. The numbers mirror the intended cost hierarchy underlying the SessionGraphs approach: users switched more between session visualizations than between different tasks.

In addition, only 85 of the 248 SG-sessions, i.e. roughly a third, were manually created. Two thirds were automatically created by the system whenever it started or in the other situations listed on page 116. The participants mentioned in the interview that they appreciated most of the cases when the system created a new session. Some situations occurred, in which a user wanted to continue the last session when starting the system so he had to navigate there because the system provided a new empty session. This suggests that the approach should be modified so that it does not start with an empty session, but provides the last three sessions in session view tabs. The modified new behavior could be, for example, that a session becomes immediately the “current”, active session as soon as the user interacts with one of the visualizations in the SessionGraphs interface, whereas an empty session is created if he or she loads a page with using the web browser’s interface first.

Participants frequently deleted and renamed SG-sessions (mean: 3.5%, 121 times in total, min: 4, max: 47), more often than they renamed a single node (mean: 1%, 34 times in total, min: 0, max: 21). They used the find function (1%, 39 times in total – min: 0, max: 19) to search the SessionGraphs history by querying it. They made use of the drag-and-drop functionality provided by the task chooser (26 times – min: 1, max: 17), and they actively created new SG-tasks (21 times) or renamed existing ones (20 times). The high variance in this value vividly expresses the different attitudes of the participants: some were quite willing to invest additional efforts to gain more benefits from the tool, whereas one left the automatically created task hierarchy practically untouched.

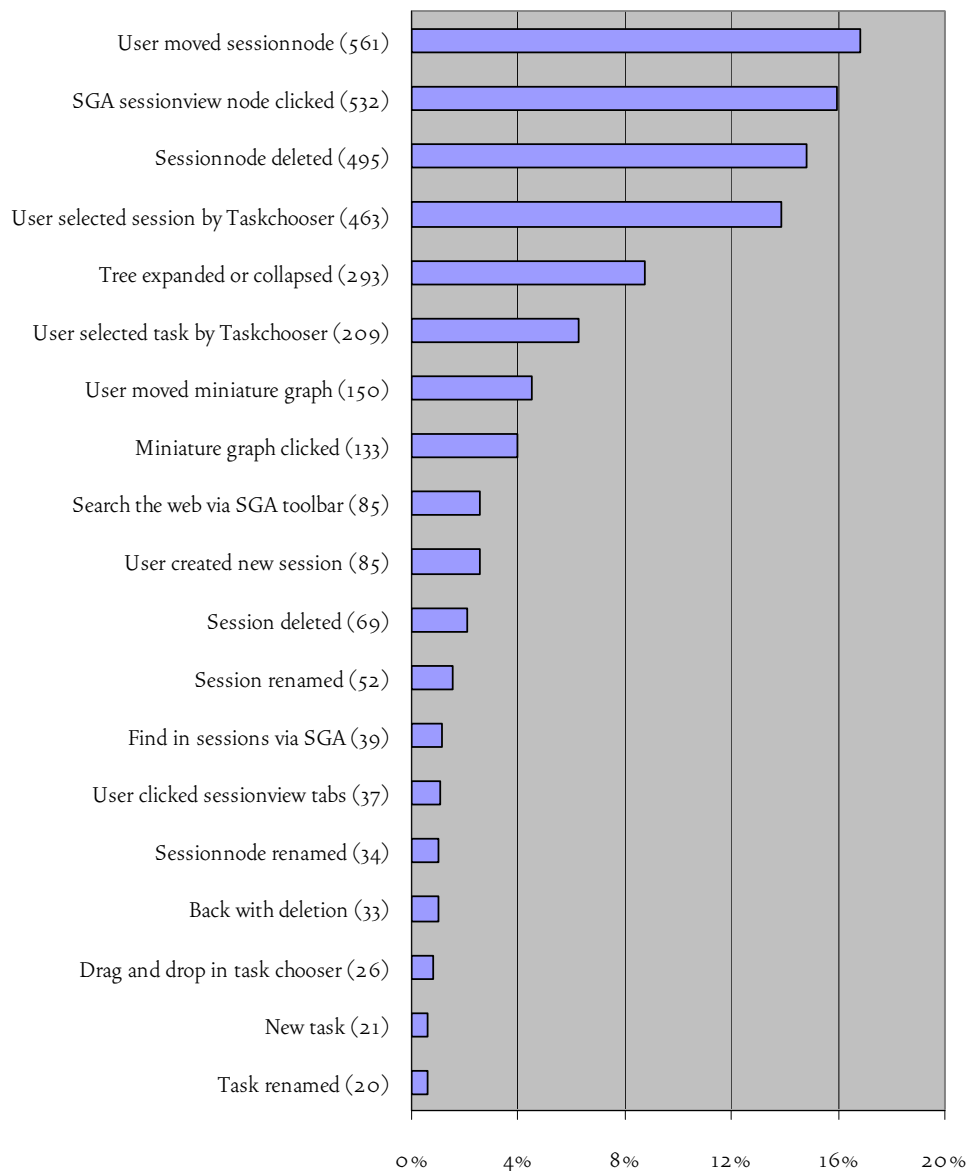


Figure 4-9: SessionGraphs interaction techniques as applied by the subjects. Numbers in brackets represent absolute occurrences as sum of all subjects. SGA = SessionGraphs Application.

The function “mailing a session to a friend” was only used 4 times in total for test purposes since the subjects were not collaborating and there was neither a reason nor a chance of using it in a meaningful way. One participant commented, however, that he would rather send a single link than data of a whole session or task, since the colleague would have to review all the single pages again. This motivates to refine the mailing concept and to consider ways to easily guide the recipient through the received data. One possibility could be a function based on the metaphor of flipping through a book, a technique called *flying* by Lai and Manber (1991) as already

applied in the WebBook project (page 57). Moreover, the sender could be supported in creating an easily navigatable sequence of relevant pages, potentially by deleting or hiding less important pages and enriching others with personal comments.

Look-ahead graphs were successfully applied by at least one participant who mentioned this in the interview. However, they seemed to have been activated much less than expected. A missing event prevented to create quantitative statistics about this function, but participants reported that look-ahead graphs were displayed only very rarely, if at all. It seems worthwhile to take a closer look at the quality of long-term revisits along similar paths.

Concluding, all major functions of the SessionGraphs history interface were successfully applied by the participants. For most functions, the participants contributed valuable ideas about how to further improve them and to make them more usable in an actual working context.

Insights Based on Interviews & Areas of Improvement

The results of the interviews underlined the previously described findings. Four participants reported that they were able to apply the tool in a meaningful way for several relevant tasks. When asked about the tool's concept in particular, neglecting the occasional technical problems (addressed further below), they gave high ratings of the tool's abilities. Two explicitly valued the idea of graphical session visualizations as an improvement over current text-based history tools. One mentioned that he liked to watch the animated movement during surfing, and that it made his web experience "more enjoyable". Another said that he especially liked to play around with the growing, moving session visualizations while waiting for a page to complete loading or while thinking about how to proceed next with his current task. Furthermore, the slowly decreasing "liquidity" of the animated views gained positive feedback. One participant suggested to stop the animation as soon as the user interacts with the visualization. This indicated a positive quality of the SessionGraphs approach that does not itself lead directly to more effective navigation, but that adds to the subjective impression.

The possibility for manually modifying the layout was explicitly valued by four participants. Thumbnails were mentioned as being of big help. One suggested showing them in the session views instead of abstract circles. One participant, however, did not really use the session visualizations; he started the tool, opened the SessionGraphs window behind the browser windows and left it relatively untouched. When he opened the visualizations that had been created, he said he was irritated by the large number of nodes he was not able to immediately make sense of. This may suggest that the preferable use of the tool includes the possibility to see the growing visualizations all the time and not to keep them hidden or closed. A further advantage explicitly valued by three participants was that the SessionGraphs

visual history did not forget any detail over time – in contrast to the tools they knew from common web browsers.

One important goal of the study was to identify areas for further improvement. Here, the interviews turned out to be an invaluable source of hints and ideas for enhancing the existing approach. They encompass *technical* issues related to the technical setup, implementation details, and *conceptual* issues such as the chosen graph layout, and the manual maintenance of tasks.

Technically, the decision to build a standalone application, loosely connected to the web browser, turned out to be an adequate solution for these first evaluations. The benefits included browser independence and local modifiability. The choice of Jazz as a visualization toolkit and Scone as a proxy also proved to be well-suited; both served their purposes. Furthermore, the thumbnail generation and the drag and drop support also worked out well.

A further outcome, however, was that a history tool that aims at supporting users in their actual day-to-day activities should be integrated more closely into a chosen web browser. This would allow for easier window management since the history visualizations could be integrated into the browser's side bar and users would not have to switch between unrelated windows to use the visualizations. Moreover, it would allow the following two technical shortcomings that were encountered in this study to be better addressed.

The first one relates to the **AppletConnector** solution, the invisible applet integrated by Scone into every visited HTML page. The integration in SessionGraphs was not as reliable as expected due in part to the interfering JavaScript code in several web pages, as well as to an unreliable applet viewer that crashed from time to time, resulting in situations where the applet was not running and the related functions were missing. This impeded the reliable control and addressing of specific browser windows. When the applet existed the page was loaded to the correct window, otherwise, the **BrowserControl** class opened the page in a new window. This led to the transient feeling of lost control, which turned out to be detrimental in an evaluation.

The second technical problem was caused by undetected frame pages. This occurred, for instance, when the frameset was not created by plain HTML code, but by JavaScript commands. Such frame pages could not be recognized by the SessionGraphs prototype. As a result, the session visualizations became cluttered – a single frame page would be represented as several visual nodes.

In addition, several shortcomings of more limited scope have been reported in which the prototype showed unexpected behavior. They include problems of the Scone parser with cyrillic pages and several pages on eBay that resulted in a few crashes of the system or unsaved session histories.

The major technical problems should be able to be circumvented in future projects by the closer integration of the prototype's code into the browser's imple-

mentation. This would allow observing inter-window and -tab navigation more accurately and accessing the browser's internal representation of frame pages even if created by JavaScript.

The participants' reactions to these technical shortcomings revealed a general issue concerning longitudinal studies that analyze a prototype's usability under naturalistic conditions: users seem to expect product level quality in terms of usability and reliability, comparable to Internet Explorer that looks back on three thousand man-years of development or yearly development costs of 100 million US\$ (as estimated by Orłowski 2002). In contrast to temporally constrained, controlled studies, people are much less willing to make compromises when using a tool during their daily work. Even very small or casual shortcomings that made the user feel uncomfortable resulted in his switching back to a less sophisticated but more stable solution. Such expectations, as justifiable as they are, can hardly be met in a project of comparative limited scope as this thesis. This should be taken into account for future research projects, and when interpreting the results obtained in this study.

Regarding *conceptual problems*, two main issues arised. The first problem concerned the automatic layout of session visualizations, which require further improvement; the second referred to the additional overhead of manually organizing the sessions and tasks.

One user with a 1024 x 768 pixel screen, the lowest recommended resolution to use the prototype with, reported problems with cluttered views after long SG-sessions. Two others said they would prefer more spiral-like figures that make better use of the available visual area, and two wanted to be able to set and delete edges manually as they liked. Thus, to enlarge the group of users who are pleased and well supported by the session layout, a beneficial next step would be to re-design the layout algorithm and to increase the possible ways of interaction.

The manual organization of sessions and tasks received mixed feedback. One mentioned that it worked well for him – he mostly manipulated the session views according to his needs immediately during his browsing activities, which he said did not bother him. Sometimes, at the end of an SG-session, he would also delete nodes that he considered of little or no interest for the future. According to two other participants, making manual task organization easier would be a large improvement as the necessary additional overhead at present was too large. One mentioned that his default SG-task “temp” was becoming crowded after several days, when he did not manually reorganize its tasks and sessions.

An additional improvement would be to make it easier to retrieve history information from the task hierarchy. One said that it was easier for him to re-find a page using Google than by searching the task hierarchy. Thus, if the prototype could provide a way to search the local history in a similar way to how easily Google searches the web, it would be a great advantage for all users. The local history would then offer an important benefit due to its much smaller search space. A promising

approach for improved session and task organization will be presented in chapter 6.1.2.

4.3.4 Conclusion

Four out of five participants were able to successfully apply the prototype for their day-to-day activities, both for short and long-term revisitations. They used SG-sessions as a construct, they created new ones, switched between parallel ones, and resumed earlier ones during long-term revisitation. The created session graphs and the response of the participants showed that the applied algorithms produced meaningful, usable session visualizations. Four participants considered the graph-based visualization more helpful than the currently available history tools; one of them said that it outperforms all text-based history tools he was aware of, since it supports his memory much better than textual history tools do. They appreciated that the system did not define SG-session borders too restrictively and that they were able to define the borders themselves. The measured numbers of interactions show that they actually used the session and task visualizations as intended by the underlying cost hierarchy, i.e. they used the session views more than the task views and these more than the task chooser.

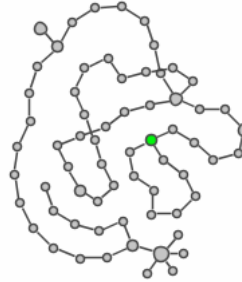
The study also revealed substantial technical and conceptual shortcomings that should be addressed in future versions of the tool and in comparable history projects. Technical shortcomings comprised problems with frame pages (“multiple-node”-representations of single frame pages) and the use of multiple windows (broken window references). Smaller, occasional technical obstacles (e.g. with JavaScript-enhanced web sites like eBay) led to the perception of reduced system reliability and substantially influenced several participants’ willingness to use the prototype continuously.

On a conceptual level, two issues should be addressed in the future: first, the tasks approach should be reconsidered as only a few users actively used it to organize their session data. Since the approach is conceptually based on recording every single session, no matter how short, the user is either confronted with a substantial amount of necessary manual re-organization of sessions into meaningful tasks, or with a “temp”-task that would soon contain too many sessions to be easy to handle. Thus, a new way to deal with the large amount of sessions should be found (one suggestion is presented in chapter 6.1.2). Second, some participants mentioned that there were situations in which the session views made inefficient use of space. Especially when navigating long sequences of links without considerable backtracking, space-consuming “zig-zag” shapes emerged, based on the current layout algorithm, that were hard to differentiate. Future versions of SessionGraphs should be able to create “denser” visualizations, i.e. to show more session details within the same

screen space. Two subjects recommended a more spiral-like session layout (this will be focused on in chapter 6.1.1).

By incorporating the solutions to these issues, a more improved version of the prototype could be developed. However, the shortcomings addressed above also showed that several assumptions of what people actually do on the web might have been too coarse, outdated, and even misleading. A better understanding of the users' actual behavior is vital in order to precisely address their needs. In particular, open questions remained about how users exactly shape their sessions, what techniques they use to visit and revisit pages in the short term, and what role search engines and multiple windows / tabs play in this context. Concerning long term revisits, it was still unclear why look-ahead graphs were so infrequently used, whether people actually revisit groups of pages in the long term, and what problems occur when trying to re-find single specific documents.

Thus, in order to gain new insights into the requirements for the next generation of web history tools, a more in-depth study seemed necessary, taking a closer look at how people actually use the current web– with their regular, unassisted browser – and how they go about revisiting pages today.



5

Re-Studying Re-Visitation

*Human beings, vegetables, or comic dust,
we all dance to a mysterious tune,
intoned in the distance by an invisible player.
(Albert Einstein)*

Together with three colleagues, I designed and conducted an extensive long-term clickstream study on how people actually used the web between late 2004 and spring 2005. The study focused on various details of browser and web usage with a special focus on revisitation. It was designed to re-question and update existing assumptions, as well as to provide further insights into how the SessionGraphs concept could still be improved to match requirements of the contemporary web user based on actual findings about current web navigation behavior.

5.1 Reasons For Re-Studying

A new study was necessary for three main reasons, as elaborated below: (1) prior insights were extremely outdated; (2) existing data were not detailed enough; and, (3) sessions had not been analyzed in detail yet.

Regarding the first aspect, before 2005, most insights on revisitation were based on the three comprehensive longitudinal clickstream studies already presented in chapter 2.2.5 (Catledge & Pitkow 1995; Tauscher & Greenberg 1997a;

Cockburn & McKenzie 2001). In 2005, these data could rightfully be considered out of date. Data had been collected 5, 9, and 10 years earlier. Since then the web has changed drastically according to the kind and amount of delivered content, the number and background of users, and the technologies used for page encoding, viewing and transmission (see chapters 2.1.1 and 9.2). We expected that the modified technology, content and experience, also would significantly influence the behavior and tasks of users.

Regarding the second aspect, existing data did not provide enough detail to answer all the questions of interest. More detailed knowledge was necessary to reconstruct and analyze the user's single interactions with the browser and low-level navigation patterns. Not much was known about session-based behavior. The most recent study only recorded daily history files – giving no possibility of analyzing the exact sequence of visits or sessions. Furthermore, no details were available about long-term revisitations – like revisits to specific data pages for further elaboration after several weeks – that still may be important to the user. Such revisits do not attract attention in common statistical analyses. Users have to be asked about them and log files have to be analyzed in specific ways. Neither was done in earlier studies. Consequently, no thorough analysis was conducted of the structure of single sessions and their relation to revisits.

These reasons motivated a new, more detailed study. Our case study involves four researchers' work over two years and goes beyond previous studies both in size of collected data and in sample broadness. The individual contributions of the different authors are identified in appendix 9.6. In the following, only those results will be presented that are of interest in the context of SessionGraphs. Our publications inform about further results and detail (Obendorf et al. 2007; Weinreich, Obendorf, Mayer & Herder 2006c; Weinreich et al. 2006b; Herder, Weinreich, Obendorf & Mayer 2006; Weinreich, Obendorf, Herder & Mayer 2007).

5.2 Method, Participants, Preprocessing

Methodically, a combination of two interviews (one at the beginning, one at the end) and client-side logs of the web browser's clickstreams were chosen (cf. chapter 4.1) – similar to the longitudinal SessionGraphs study – with the addition however that they were performed in combination with an unmodified browser. This allowed both for measuring detailed navigation activities, as well as gaining insights into the participants' subjective intentions.

Twenty-five unpaid German and Dutch volunteers participated in the study. Six of them were female (24%). Three of the Germans were located in Ireland or

New Zealand. Ages ranged from 24 to 52 years (mean: 30.5), web experience from 3 to 12 years (mean: 8 years). Sixteen participants (64%) worked in computer science whereas the other nine (36%) came from a different background, two of which worked in psychology, otherwise, one each in sociology, geology, electrical engineering, trading, coaching, history, and photography. Seventeen were PhD students, six were in their professional lives and two were students. Neither the computer-science attribute nor gender produced measurable effects. Seven additional candidates were unable to complete the study due to personal or technical reasons and were therefore not included. Interviews were conducted in the participants' native language.

Due to the relatively small sample and the computer science (CS) bias, the results of this study should not be over-generalized. The CS bias, however, shared with the three earlier studies, increases the inter-study comparability with them. Similar to the SessionGraphs study, recruiting participants was difficult due to the high degree of trust necessary for sharing browser related activities. A big advantage was that no additional installation was necessary and no new interface had to be learned. The participants were free to temporarily turn off the logging e.g. for privacy issues or to remove sensitive items from their logs before submitting them. Most participants were logged at their work place. Thirteen accessed the web also from home using an unlogged browser. This influences the data, but cannot be avoided in a naturalistic long-term study.

Logging technology was again based on Scone (Weinreich et al. 2003). JavaScript code was added to every page to gain detailed information about navigation actions and page characteristics.¹ The Scone plugin "WebChains" was designed to generate the log files. It already filtered out several unwanted events.

Intense preprocessing of the logs was still necessary to gain entries that represent single, user-initiated events. Before the cleaning process, the data comprised more than 160,000 page load events collected by the Scone software and nearly 150,000 user interface events collected with the modified Firefox browser, which also included non-navigation interface interactions such as tabs selection events. Each event had up to 27 parameters. After cleaning, 137,272 page load events remained. A large number of 'polluting artifacts' – 33% of the primary entries of users who did not use an ad blocker – had been removed: inline frames, sub-frames of HTML framesets, advertisements, pop-up screens and automatically refreshed pages. The cleaning process is precisely described in (Weinreich et al. 2006a). Comparably extensive and careful preprocessing was not reported by the authors of

¹ Fifteen participants were equipped with a modified version of the open-source browser Firefox 1.0 (Mozilla__Project 2007). This enabled us to collect detailed information about user interface events – also such that do not lead to a page visit. Ten users continued using their familiar browser. Thus, detailed visit-logs were obtained for all 25 users and additional interface-interaction-logs for 15 users. These more detailed logs were used to develop a heuristic to automatically tag all 25 visit-logs with further information, e.g. back button menu clicks (more information in Weinreich, Obendorf & Herder 2006a).

the three previous studies – although it should be noted that many of the causes which created the artifacts did not yet exist.

Still, our study can be compared with the earlier ones (see Table 5-1). With 25 participants this study is reasonable in size; the average of 105 logged days per user is comparatively high (between two and seven months per participant); with almost 140,000 entries this study comprises by far the most entries amongst all four studies – even after preprocessing; our study consists of 7.0-, 4.4-, and 1.6-times as many visits compared to the earlier studies.

One of the most apparent results was the declined percentage of back events that was only about half as high as in earlier studies; likewise, the average recurrence rate was comparatively low with 49%. Individual users visited between 25 and 284 pages per active day on average, which already reflects the individual differences in web usage. The mean of 90 visits per active day is much higher compared to the means found in earlier studies.

5.2.1 Detection of Technical Sessions

The following terminology refers to the nomenclature introduced on page 27. We applied the most commonly used heuristics for the detection of *technical sessions* in client-side log files to our logs – a timeout introduced by Catledge and Pitkow (1995). Their algorithm detects session borders by looking for an interval subsequent to a visit of at least 25.5 minutes, in which no other visit occurs. The first visit after such an interval indicates the begin of a new T-session. This procedure was used for the analyses following below. The choice of this procedure will be discussed more thoroughly on page 182.

5.3 Results

5.3.1 General Measures

Table 5-2 shows general measures derived from the cleaned data. Both the number of total visits and the individual recurrence rates vary highly between the subjects. Between 912 and 30756 web pages were visited per subject (mean: 5491) with individual recurrence rates between 20% and 72% (mean: 49.0%, $\sigma = 11.3\%$). The high number of visits for all users meant that all recurrence rates could be considered valid – Tauscher showed that recurrence rates stabilize already after about 200 visits (1996a: 39). The high variances also have provided evidence for different styles of web usage and / or different kinds of tasks.

	This Study	McKenzie & Cockburn ¹	Tauscher & Greenberg ²	Catledge & Pitkow ³
Time of Data Collection	2004–2005	1999–2000	1995–1996	1994
Applied Technology	Intermediary & Firefox 1.0	Netscape 4.x: history.dat-files	Modified X-Mosaic 2.6	Modified X-Mosaic 2.4
Data Captured	Visits, User Actions, Page Inform. ⁴	Number of Visits as summarized in history.dat files	Visits, User Actions	Visits, User Actions
Number of Subjects	25	17	23	107
Background of Subjects CS = computer science	64% CS, 36% others	100% CS	100% CS	100% CS
Logged Days per User	Ø 105⁵ (52-196)⁷	119	about 42 ⁶	21
Total Visits	137,272⁷	84,841	about 19,000 ⁸	31,134
Total Distinct URIs	61,041	17,242	– ⁹	– ⁹
Visits / User	912-30,756 Ø 5491¹⁰	281-23,973	between 300 and ~3300	Ø 291
Visits / Day	Ø 47¹⁰	Ø ~42	Ø ~20	Ø ~14
Mean Recurrence Rate	49%¹¹	(81%) ¹²	58%	61%
Navigation via Link-Anchors	44%	– ¹³	42%	52%
Back Events	14%	– ¹³	30%	41%
URL Entry / Forward Button	4% / 1%	– ¹³	7% / 1%	unmeasured / 2%
Bookmark Usage / History List	2% / 0.2%	– ¹³	3% / 1%	2% / 0.1%

Table 5-1: Our study's results compared with the three major earlier ones.

¹ (McKenzie & Cockburn 2001; Cockburn & McKenzie 2001)

² (Tauscher 1996a; Tauscher & Greenberg 1997a)

³ (Catledge & Pitkow 1995)

⁴ E.g. num. of links (int. & ext.), window size, click pos., stay duration, get & post params.

⁵ We measured a mean of 54 (24-147) *active days* – days with at least one visit.

⁶ “Subjects used Mosaic for approximately 6 weeks.” (Tauscher 1996a: 38).

⁷ After data cleaning – removal of artifacts by frame pages, ad services and auto-reloads.

⁸ Ø 910.87 entries, incl. 90.46% nav. events (Tauscher 1996a: 53) × 23 = 18,951.

⁹ Can be derived from recurrence rate and total visits.

¹⁰ A mean of 90 *visits* was found in our study *per active day*.

¹¹ Rec. Rate based on full URI, Get-params, no Post-params. Also frame sets considered.

¹² Rec. rate based on URI *without get- and post params* (McKenzie & Cockburn 2001: paragraph 3.1). Thus, this value is *not* comparable with other rates. If we ignore get- and post-params we receive 69.4% for our data (compare page 176).

¹³ Not measurable, since only the daily history.dat log files were used for analysis.

The limited number of subjects and the high variances recommend not over-generalizing the results – neither in this case nor in the three earlier studies. The differences between individual measures seem to be much more expressive and valuable for future designs than the derived statistical means.

Looking closer at the recurrence rate, we find that this rate highly depends on its definition and on the chosen preprocessing. An according differentiation was not accomplished by earlier studies. To analyze the effects of different definitions we used the cleaned log files and considered only visits to non-frame-pages to keep the results as clean as possible. The resulting mean recurrence rate was 47% (compared to 49% with frame pages considered). Before data cleaning it was 54.1%; individual rates were 6% to 20% higher than afterwards. These recurrence rates were based on a definition that included HTTP GET parameters following the question mark in URIs. If we had defined the recurrence rate (as Cockburn and McKenzie did) based on URIs that were truncated after the question mark, then our mean recurrence rate would have been as high as 69.4%. If we, however, include also POST parameters in the definition, it is reduced to 43.7%. Finally, if we only consider revisits to pages containing the same content (accomplished by comparing fingerprints calculated from the pages' content), we gain a recurrence rate as low as 34.6%.

None of the presented definitions for a recurrence rate can claim to be the only appropriate one. Each study should decide and then publish the exact way used to calculate recurrence rates. Only then, the results can be compared.

The large diversity in web-related habits became even more obvious when comparing the different users' behaviors on a daily basis. Already Tauscher et al. reported that web-related activities do not occur equally distributed over time, but instead periods of intense use alternate with phases of inactivity (Tauscher & Greenberg 1997a). To find out more about this diversity, we plotted all the web activities of our participants onto two dimensions, the result of which can be seen in

	mean	median	min	max	stdev
Total Visits	5490.9	3784.0	912	30756	6564.8
Recurrence Rate (Revisits / Visits)	49.0%	47.3%	22.4%	78.8%	11.3%
Total Weeks of Study	15.4	16.0	8	28	4.6
ActiveWeeks of Study	12.2	11.0	6	25	4.6
Mean # of Revisits / Active Week	203.9	143.3	46.8	766.1	173.2
Mean # of T-Sessions / Active Week	14.7	12.6	4.9	41.4	8.3
Mean # of Entries / T-Session	28.3	23.7	9.2	117.2	21.1

Table 5-2: General measures of web usage study with common browser.

Figure 5-1. It shows exemplary clippings of six users' web activities, each comprising 120 consecutive days represented on the horizontal axis. Black dots indicate activities in the vertically represented interval of 24 hours. Subjects A & B were logged both at home and at work - with obviously different sleeping patterns, subject C and D were logged at work and said they used the web only for work. Subject E used a laptop both for private and work issues; F was only logged at home during private usage of the web.

The visualizations reveal that some participants (D, A) used the web for longer coherent activities within a day than others (E, F). The mean number of technical sessions per day was 2.4 for D (average session length: 91 minutes), and 6.3 for A (avg. length 31.4 minutes) while they were 1.4 for E (26.5 minutes) and 1.8 for F

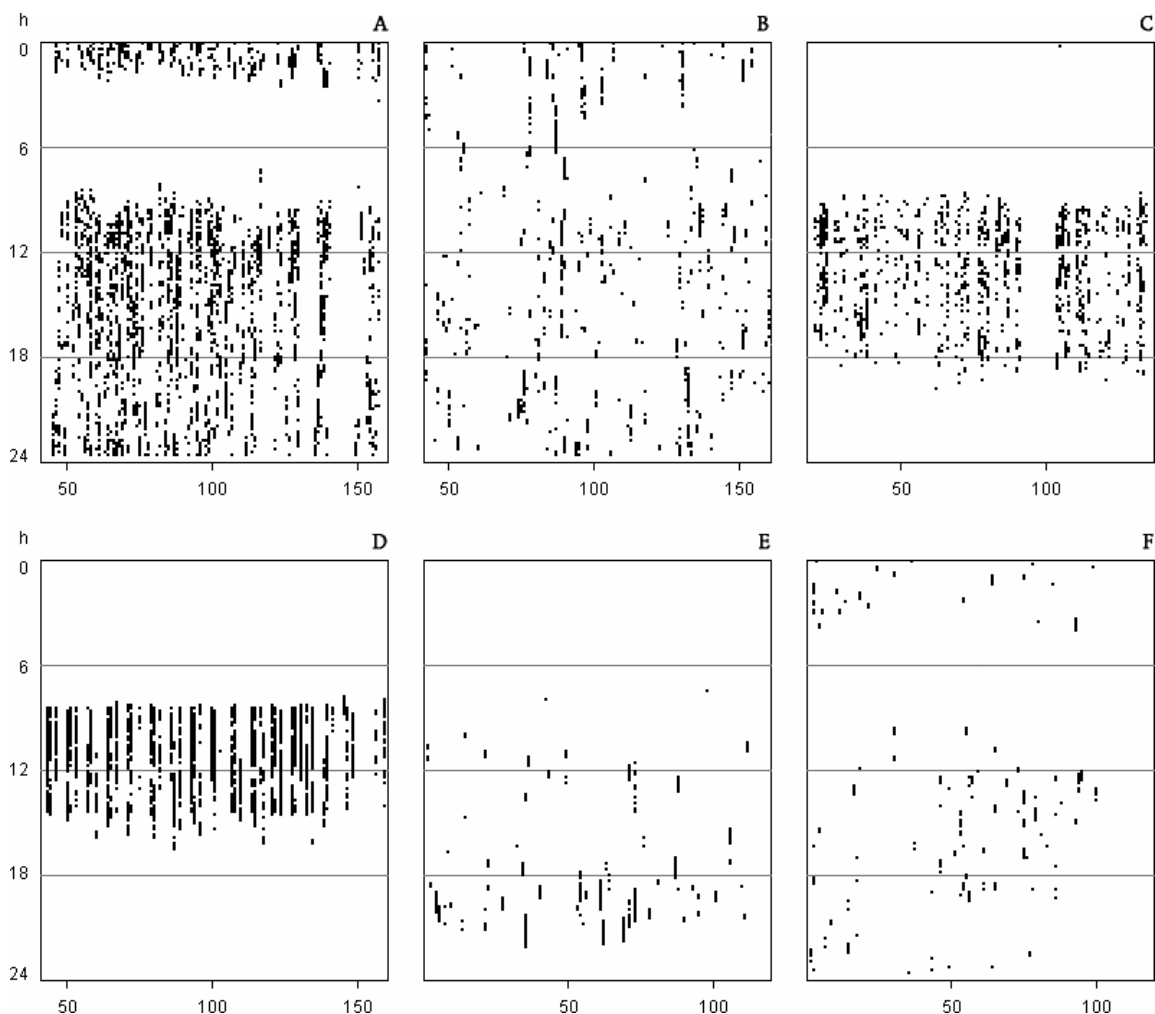


Figure 5-1: Web activities as distributed over the day and over weeks.

Six exemplary subjects. Horizontal scale: sequence of 120 days, vertical: time of day, resolution: ten minutes. A black dot represents one or more load events within the respective ten minutes' slot. The following values represent (mean session duration in minutes / mean number of sessions per active day): A (31.4 / 6.3), B (18.5 / 2.8), C (17.9 / 5.0), D (91.0 / 2.4), E (26.5 / 1.4), F (12.6 / 1.8).

(12.6 minutes). Apparently, some participants (C, D) have more constant usage patterns than others (B, F).

Furthermore, the absolute duration spent on the web per active day shows high individual variance (see Figure 5-2). Subject A and D spent by far the most time per day on the web, around three and a half hours. E and F represent the other extreme, both under one hour.

A manual categorization of the most frequently visited 300 domains (of about 10,000 visited domains in total) combined with questions in the first interview revealed the major kinds of uses for which our subjects used the web during the study (see Figure 5-3). They comprise work and university, tools and services, search, personal and recreation, commerce and trading, news, travel, and other. The visits to these domains add up to about 100,000 visits, roughly two thirds of all logged visits. This shows that the web was used by the study participants primarily for work related issues and for search, followed by news reading, commercial activities, and other services. Personal topics, recreation, and travel form the lower end of the list.

The high intersubjective variances mentioned above suggest that future history tools should either be capable of absorbing these variances conceptually or be adaptable by the users to their individual needs. The SessionGraphs concept is well prepared for these variations. It introduces the concepts of sessions and tasks that are compliant with the most different types of behaviors. It facilitates sessions of different size, duration, and frequency. Additionally, the suspended mode offers the possibility to not record specific phases, as desired by one participant.

5.3.2 Less Back, More Windows, Much Google

Even if the results varied highly between the subjects, three generally measurable tendencies could be identified. They comprise the relatively low usage of the back button, the high use of multiple windows and tabs, and the high percentage of Google visits.

The use of the back button only accounted for 14.3% of all interface actions that led to a visit (compare also Obendorf et al. 2007). Earlier studies revealed values more than twice as high (Catledge & Pitkow 1995; Tauscher 1996a). This indicates a potential change in short-term revisitation behavior. Two other interface events occurred more often than in earlier studies: The act of loading a page into a new window or tab now accounted for 11% of all events, the submission of forms for 15%. Both may be partly responsible for the lower share of back usage as explained below.

The low back button rate was not caused by an increased use of the back button pull-down menu, as we first assumed – only 3% of all back button events indi-

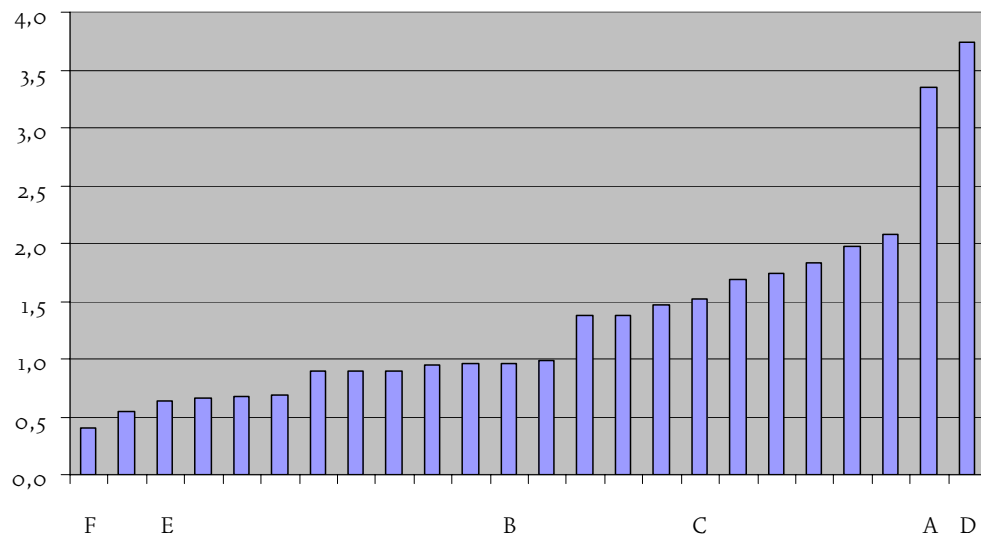


Figure 5-2: Daily time spent in technical web sessions.
Per active day and participant. Ordinate: time in hours, abscissa: the 25 participants.

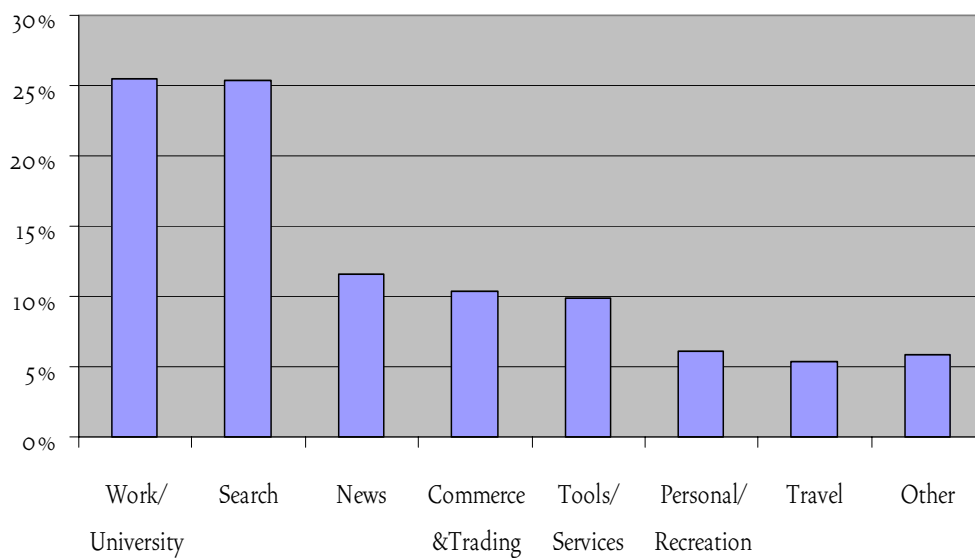


Figure 5-3: Classification of the 300 most frequently visited domains.

cated the use of this pull-down menu. Rather, two additional shifts in web browser usage seem to be responsible.

First, we found evidence that many hub-and-spoke navigations have been replaced by navigation between multiple windows and tabs. Already Aula et al. (2005) found in their summer 2004 survey involving 236 experienced web users that they “almost always” use multiple tabs and “often” use multiple windows (median values). In our study, we found that those participants who created the top third of new

window events used the back button less (10.2%) than those who created the bottom third of events (16.4%, $t=2.509$, $p=0.026$). Six of the fifteen Firefox users reported to make frequent use of browser tabs.¹ They backtracked less often (9.9%) than the remaining users (18.3%) that hardly opened any tabs ($t=2.311$, $p=0.038$). They reported to utilize tabs as a means to compare pages or to keep important information at hand. Many of our participants applied a modified short-term revisitation technique – instead of using backtracking in one window to explore different spokes of a certain hub, the hub page was now opened in one window while several spokes were opened in other simultaneously opened windows or tabs. The interviews revealed that this behavior was preferred due to faster back and forth navigation and the opportunity of comparing pages more easily. The result is a considerable replacement of backtracking activity by inter-window navigation. A question that arises is how studies like ours and history tools like SessionGraphs should deal with such behavior. It can rightfully be argued that navigating back and forth between different windows and tabs is another form of backtracking. It is, however, not measured in our setup or earlier ones since it does not create a new page visit event. Still, future history tools will have to provide adequate means of supporting this kind of inter-window navigation. Current back button concepts, with their separate, window-based stacks of back events, are unable to fulfill these needs. Furthermore, current strategies such as Windows bar items and tab navigation seem only to be a first step in the right direction of inter-window navigation.

The second reason for the lower use of the back button seems to be the web's shift from an originally static character to an increasingly interactive one. This becomes evident, for example, by the increased number of form submissions. Frequent form users in our study did not use the back button as often (the top third of 'form submitters': 9.2%) as the remaining participants (16.2%, $t=2.715$, $p=0.012$), a marginally significant result. Also in this area, history tools will have to be adapted to the modified needs. Backtracking in form based environments today often introduces problems such as error messages or lost content. Many web applications even completely disable the back button and provide their own means of getting back or, more often, some kind of undo. Future history systems should try to achieve a more consistent way of treating back in form environments. This, however, will only be possible with a unified behavior on server-side.

The third striking result was the high use of Google – for some participants, at least (see Table 5-3 and Herder 2006: 17off). Between 28.0% and 93.9% of our users' T-sessions involved at least one visit to a Google page (mean: 56.6%, $\sigma=20.2\%$). Per user 4.0% to 39.3% of all visits were Google visits (mean: 16.9%, $\sigma=8.6\%$). Google revisits show a comparable recurrence rate (Google revisits / Google

¹ Tab usage like open, select, and close averaged to 19.2% of all UI actions.

visits) of 42.5% to the overall recurrence rate, also with high individual differences (min: 12.3%, max: 88.7%).

Several subjects mentioned that they use Google as an equal history technique to browser based history techniques. Others mentioned it to be their work-around when all the other mechanisms fail. Still, Google is not an ideal long-term revisitation tool. Besides the deep web issue mentioned earlier, an obvious advantage of history systems compared to web wide search engines is that the search space is by far smaller. Thus, it would be a first important improvement, if browser based history systems became as “easily searchable” as the web in combination with Google, for example. Recent developments may change this situation, since Google, for example, transformed its project “Search History”, which allowed registered users to access their earlier searches, to “Web History”, which allows them to access their complete earlier history (Google 2006). Since these complete personal histories are stored on Google servers it is a highly debatable procedure in terms of privacy issues. Another project that aims at blurring the border between pages on the web and visited pages in the user’s history is HCB (History Centric Browsing – Shirai, Yamamoto & Nakakoji 2006).

In the interviews participants reported problems such as forgotten and misspelled queries, the inability to relocate a target in a large list of search results, the inability to (re-)find appropriate query parameters, or, that Google does not list the desired page at all (in one case due to robot exclusion on a forum’s site, in another case since the page was unlinked and only published by email). One way to fight these problems is addressed as mentioned above by Google’s “Web History” that collects earlier searches. Another way to support searches would be to improve the SessionGraphs approach by addressing the visualization of search engine usage in a special way (compare chapter 6.1.4).

The study revealed also interesting insights corresponding to the quality of revisits. According to the temporal character of revisits we found that short-term

	mean	median	min	max	stdev
Google Visits / All Visits	16.9%	16.2%	4.0%	39.3%	8.6%
Google Revisits / All Revisits	15.1%	14.0%	2.1%	42.8%	10.4%
Google Sessions / All Sessions	56.6%	51.6%	28.0%	93.9%	20.2%
Mean % of Google Visits / Session	17.8%	14.5%	4.3%	46.4%	10.6%
Google Revisits / Google Visits	42.5%	43.9%	12.3%	88.7%	18.7%

Table 5-3: Analysis of visits and revisits to Google pages.

All entries with the string “google” in their URL were considered a Google visit here. A session is considered a Google session when it includes at least one Google visit.

revisits (72.6% of all revisits) mostly comprised *backtracking* and *undo*-activities. Medium-term revisits (12%) often comprised revisits to frequently visited sites where the user *re-utilized* or *observed* pages. Very long-term revisits (7.6%) regularly encompassed *rediscover* activities in which the user tried to revisit one or several important pages – which are not frequently revisited. The period between one day and one week (long-term, 7.8%) was predominantly occupied by a composition of activities common in the medium and the very long-term, i.e. we found *re-utilization*, *observation*, *rediscovery*.

5.3.3 Sessions

Discussion of the Technical Session Border Detection Algorithm

Catledge and Pitkow revealed their interval by taking the mean distance between two user interface events (9.4 minutes in their data) and adding 1.5 standard deviations resulting in 25.5 minutes. The pure fact, however, that their algorithm was widely considered for technical session detection (Baldi, Frasconi & Smyth 2003; Cooley, Mobasher & Srivastava 1999; Spiliopoulou et al. 2003; Pierrakos, Paliouras, Papatheodorou & Spyropoulos 2003) is not sufficient to justify its application. In an email conversation with us they motivated this proceeding by arguing it was a standard statistical procedure to remove outliers from a dataset. This, however, is only valid for normal distributions and not for power law distributions as present in this case. It is surprising that their procedure still was applied *unquestioned* so often. One reason that no questions came up might be the fact that changing the timeout to 20 or 30 minutes would not change too much, since all these cutting points belong to the distribution's long tail.

To informally evaluate whether this proceeding actually produces reasonable results, one of our subjects looked through about four months of his web-related activities based on his log file. He manually assigned the borders of 292 intentional sessions plus 6 intentional sessions that occurred concurrently to other sessions. Considering the 292 subsequent intentional sessions, the following results were found.

The probability that the next event in the log file starts a new intentional session – and does not belong to the current one – grew with increasing intervals of inactivity since the last event. And, in fact, the probability that a new visit represents the beginning of a new intentional session started exceeding a threshold of 50% at about 25 minutes in this user's log (see Figure 5-4, and also Herder 2006, who used bins of 5 minutes instead of one minute). This provides some justification for rightfully applying the 25.5 timeout.

Additional justification was found when discussing exemplary T-sessions with our participants in the second interview. The borders detected by the 25.5 minutes algorithm matched very well the intentional borders of the few selected sessions

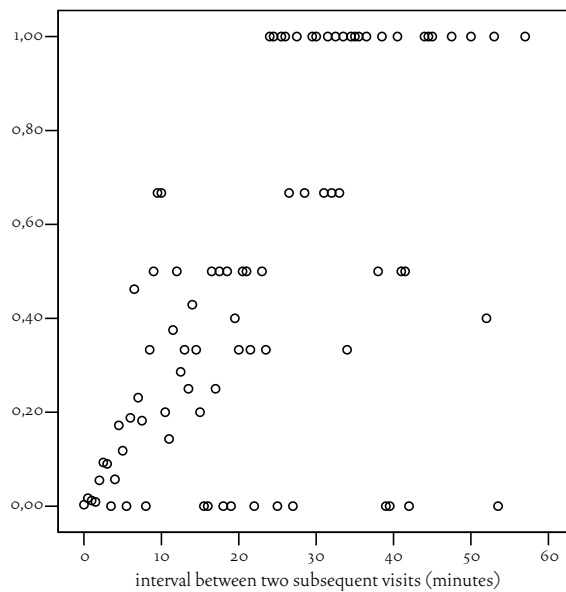


Figure 5-4: Empirical validation of 25.5 minutes session border timeout. Based on the detailed analysis of one participant's log file (four months, 292 intentional sessions). The chart depicts for intervals of inactivity between 0 and 60 minutes the probability (ordinate) that the next visit starts a new intentional session.

– as stated by the interviewed subjects. In about half of all discussed T-sessions technically and intentionally determined borders were identical. In many other cases the difference measured only a few visits on both borders.

Several alternative heuristics for session detection are presented by Spiliopoulou et al. (2003), many of them based on the site structure or referring pages. She concludes that no heuristic is the best for all possible cases and that different heuristics lead to different results.

Still, we needed some heuristic to split our log files into chunks. Given the tradition of the 25.5 minutes timeout and the informal justification presented above, it seemed reasonable to follow this tradition. When doing so, one has to keep in mind that any T-session may consist of multiple intentional sessions – or parts thereof, and that any intentional session may be split over multiple T-sessions.

The 25.5 minutes timeout could also be a first valuable step of automation of session management in the SessionGraphs approach, as demanded by some subjects of the longitudinal SessionGraphs study and discussed in chapter 5.3.3.

We asked our participants in the interviews how they think that session borders could be algorithmically detected more adequately. They mentioned that possible good indications for the beginning of a new session are the use of Google, the creation of new windows, and the visit of personal home pages. These suggestions are valuable hints for improving the SessionGraphs concept.

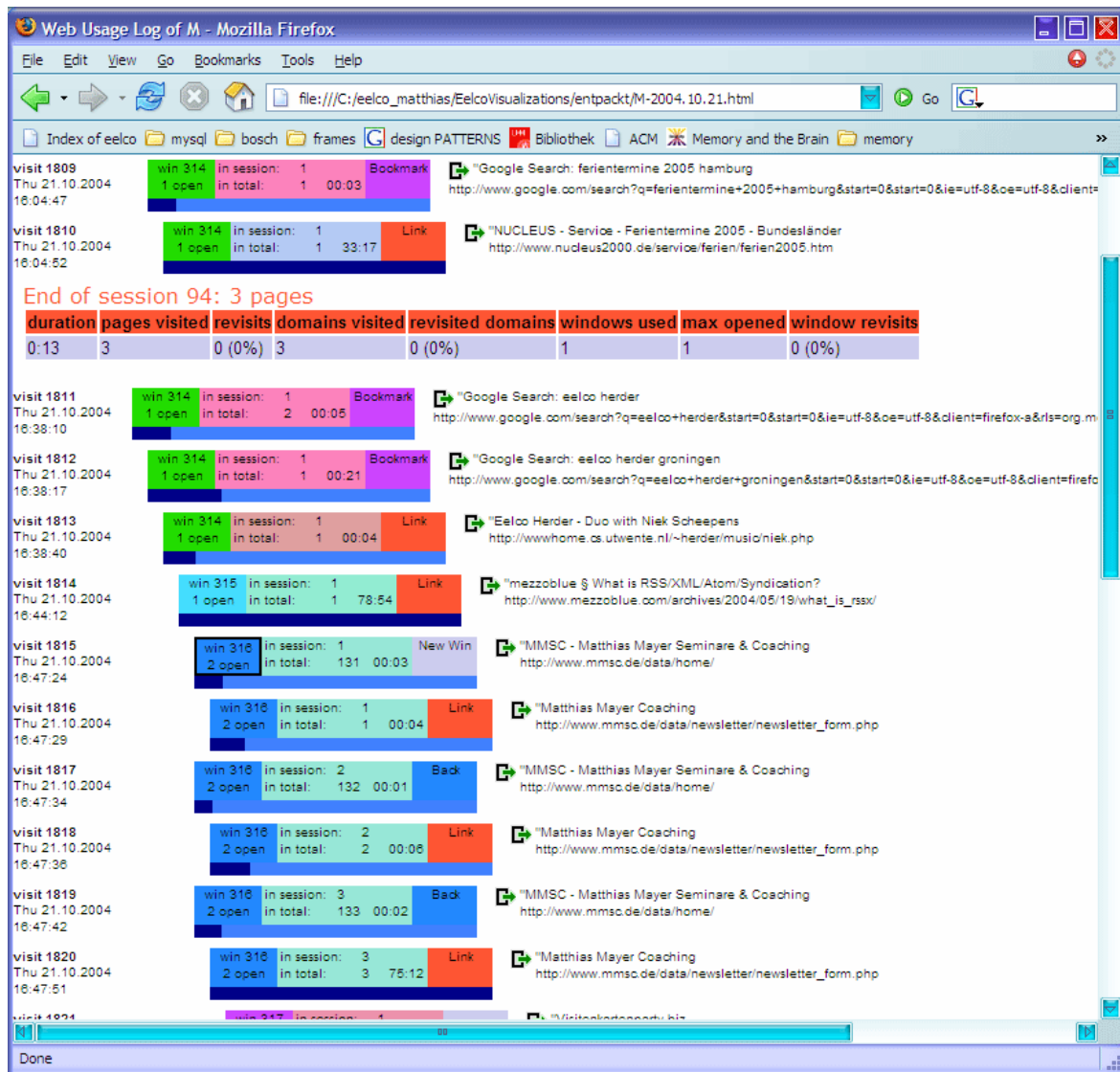


Figure 5-5: Detailed visualization of one participant's log file. This kind of session visualization was used in the interviews to discuss specific sessions and navigation behavior with the users.

Visualizing Technical Sessions for Study Purposes

To better understand the fine-grained user activities ourselves, and to discuss specific situations with our participants retrospectively, we developed a method of creating interactive visualizations of τ -sessions (see Figure 5-5 for an example). These visualizations were used in the interviews to quickly and easily remind the participants of a specific situation during the logging period and to discuss with them the single steps of their activities.

Technically, the visualization is an HTML-based color- and position-coded representation of the user's path (see also Herder 2006: 116). All τ -sessions and the visits within them are visualized in sequential order. New visits are indicated by

	mean	median	min	max	stdev
Total Number of Sessions	188	146	54	786	159
Sessions per Active Day	3.3	3.2	1.4	6.3	1.2
Visits Per Session (Avg. per User)	28.3	23.7	9.2	117.2	21.1
Distinct Pages p. Session (Avg. p. User)	18.3	15.7	5.5	71.4	12.9
Mean Session Duration (Minutes)	23.9	21.1	12.0	90.9	15.4

Table 5-4: Character of measured technical sessions (T-sessions).
Based on the 25.5 minutes timeout, over all participants.

right indentation, backtracking is indicated by left indentation. Colored boxes encode three parameters: the active window in which the visit occurred (top left box), the domain (middle box), and the load mechanism (right box). The stay-time is displayed as a bar using a logarithmic scale (bottom). Visit identifier, date, stay-time, URL and page title are textually presented. URL and title can be clicked to load the represented page. Per T-session the duration, the number of visited pages and domains, the recurrence rate, and additional information about window usage are displayed.

Results

Applying the 25.5 minutes session border timeout for all subjects resulted in the following measures (compare Table 5-4). On average, the mean number of technical sessions per active day was 3.2 ($\sigma = 1.2$) – relatively high compared to only approximately 1 in the 1994 study by Catledge and Pitkow (1995). On average, a T-session contained about 28 page visits to a mean of 18 distinct web pages ($\sigma = 13$ pages). Individual means ranged between 10 and 120 visits per session and between 5 and 71 distinct pages, indicating very different kinds of activities. The maxima of distinct pages per session ranged between 42 and 348 pages per user (mean: 138.1, $\sigma = 88$). Still, only 2.4% of all T-sessions comprised more than 100 distinct URLs. Sessions of this high order were mainly caused by two of the 25 subjects who created 32 and 43 sessions with more than 100 URLs, indicating a different, more intense and potentially faster way of using the web; eleven subjects created no sessions with more than 100 contributing URLs, for the remaining twelve subjects a maximum of six T-sessions exceeded this margin. These numbers are important for discussing the suitability of the session visualizations in SessionGraphs since they are based on constrained screen space to visualize all the nodes visited in a session (see conclusions, chapter 5.4, and the suggestion for an improved layout, chapter 6.1.1).

The mean duration of a T-session was about 24 minutes. Individual means ranged between 12 and 90 minutes. While subject D in Figure 5-1 produced an aver-

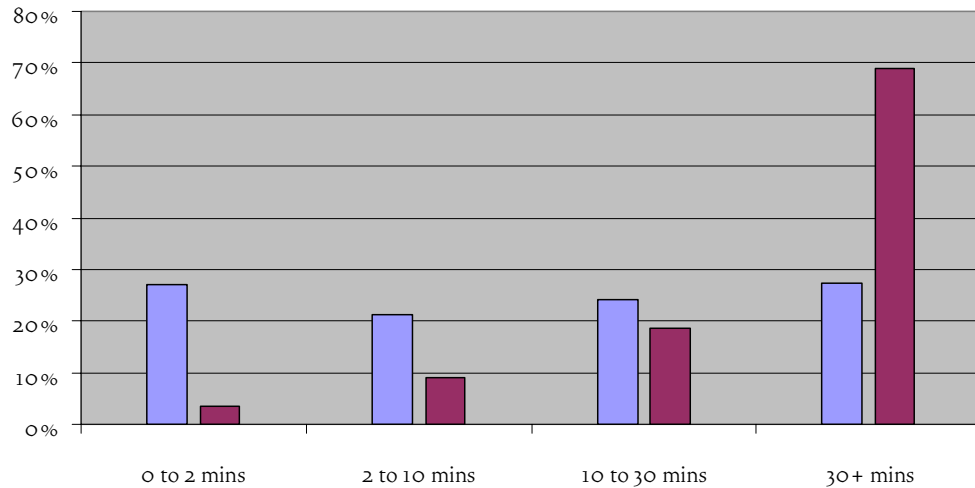


Figure 5-6: How visits are distributed over sessions of different duration. Blue (left) bars show how T-sessions of different duration account for the number of total T-sessions. Red (right) bars show the number of total visits comprised by sessions of the respective duration. In percent, measured over all subjects.

age of 2.4 T-sessions per day, each of about 1.5 hours duration, subject A created a mean of six T-sessions of about half an hour each. As already mentioned, more than half of all T-sessions contained at least one visit to a page on a Google domain (see Table 5-3).

The duration of T-sessions formed a power law distribution. Most T-sessions were very short. This suggests supporting these sessions best. The magnitude of all page visits, however, was located in T-sessions of longer duration. About a quarter of all T-sessions lasted for 2 minutes or less (“0 to 2 mins”, Figure 5-6). This quarter accounted for only about 3% of all visits. The duration of another quarter of T-sessions resided between two and ten minutes and accounted for about 9% of all visits. The next quarter lasted between ten and thirty minutes (18% of visits). The vast majority of visits, almost 70%, were created by the fourth quarter of T-sessions that lasted half an hour or longer. T-sessions longer than three hours were only conducted by nine participants and account for only about 1% of all T-sessions and 10% of all visits.

These results suggest that a session-based history tool should especially well support such sessions that last between half an hour and two hours, since they contain the vast majority of all visits.

A next step was to find out how revisitations relate to the found session structure. Here, three kinds of revisits can be distinguished (see Table 5-5): (1) pure *intrasession revisits* – i.e., revisits to pages that were visited before in the current session, but never before this session, (2) pure *intersession revisits* – i.e., revisits to pages that were visited before this session, but not before in this session, and (3) combined *inter- AND intrasession revisits* – visits to pages that were visited both in the current session and before.

		Percent of Revisits	Mean Session- length	Rel. Pos	Percent of Google Revisits	Mean Time since Last Visit
Only Intrasession	mean	46.0%	92.8	55.0	55.9%	2:56 mins
	stdev	14.1%	61.9	2.1	21.2%	1:20 mins
Only Intersession	mean	29.2%	50.0	41.1	26.1%	8.3 days
	stdev	8.8%	40.2	3.4	14.0%	3.4 days
Intra- AND Intersession	mean	24.8%	70.4	52.2	18.0%	7:12 mins
	stdev	9.3%	52.9	4.2	12.7%	3:29 mins

Table 5-5: Analysis of intra- and intersession revisits based on T-sessions.

* Rel. Pos.: Relative Position of Revisit in Session (Position 0-100).

We found for our participants that 46.0% of all revisits were pure intrasession revisits, 24.8% were both inter- and intrasession revisits, and 29.2% were only intersession revisits. This is a strong argument for the adequacy of the session-based visualizations – as for 70.8% of all revisits the user could simply stay with the current session and use it for short-term intrasession revisits. In the remaining 29% of all revisits, the SessionGraphs user would have to select the appropriate task first and switch to the desired session. When only Google revisits are considered, the share of pure intrasession revisits becomes even larger (55.9%).

Intersession revisits have a different character compared to intrasession revisits. First, intersession revisits show a trend to occur earlier in a session compared to intrasession revisits. When normalizing session-lengths artificially to a value of 100, the mean relative session position for intrasession revisits was 55.0 while it was only 41.1 for intersession revisits. Second, the mean time between two visits to a specific URL was for intrasession revisits only 2:56 minutes, for intersession revisits it was 8.2 days.

An additional insight was that the participants tended to pay visits of the same type in a row, i.e. visiting new pages they would most likely continue visiting new pages; revisiting pages from earlier sessions, they would most likely continue revisiting pages from earlier sessions; revisiting pages within a session, they would either continue doing so or visit a new page (see Herder 2006: 284). The SessionGraphs concept supports this behavior. Once the user starts revisiting pages within a session, he or she can simply reuse the current session visualization for all subsequent intrasession revisits. When pages from earlier sessions should be used, task chooser can be used to do so. When new pages are visited the focus can remain on the web browser.

One to two T-sessions have been discussed in detail with each participant in the second interview to find out more about his or her subjective motivations for certain navigation activities. Each participant was asked to locate situations where he or she “searched for important information”. To do so, the participants were introduced into the sequential log-file visualizations (see page 184). They were quite easily able to navigate in them and to find such situations. For most incidents they also were able to remember what they tried to accomplish then. If not, they were asked to look for another situation. A side-effect of this procedure was that it showed us that our participants were able to apply the term *session* very quickly and in a meaningful way to their web-related behavior, which further supports the conceptual foundation of the SessionGraphs approach.

The most common way of starting an “information search” session was to use Google – either by using the Google toolbar, Google as browser start-up page, a Google bookmark, or by entering “g [searchterm]” in the URL-entry-field. When Google was not used, often a bookmark was used or a URL was manually entered. Often, a new session was started in a new window, which suggests using such an activity to automatically provide a new SG-session.

During an information search session, the most common navigation pattern was to explore several Google search results in a hub and spoke navigation style, mostly accomplishing a breath first search along selected, promising branches. Queries were reformulated frequently to retrieve more meaningful search results (more about the formulation and re-formulation of queries can be found in Herder 2006; Aula et al. 2005). Several participants used multiple windows or tabs to store information – as an alternative for back- and forth navigation. They frequently returned to earlier search results pages search in order to explore alternative branches. Future history tools should definitely support both kinds of short-term revisitation techniques particularly well: this includes the traditional backtracking, but also “re-visitations” that are created by between-tab and -window navigations.

Our participants mentioned that they rarely started a session by using online catalogues, topically sorted hierarchies, such as Yahoo!, eBay, or Amazon. Several mentioned having done this earlier, but that they relied more and more on textual search now. Link-navigation is considered to be too time-consuming in relation to textual search – be it on a web wide basis using Google or using a dedicated search engine locally – congruent to Nielsen’s earlier findings (1997a). One subject even answered, when asked whether he used web directories: “No. I can never find what I am looking for in a web directory” – and another: “in most cases the hierarchy does not match my info need”.

The usefulness of the application of tasks is further supported by the fact that about half of all discussed T-sessions were part of an ongoing information seeking task on which the participants also worked before or after the discussed session.

This documents the existence of enduring tasks that are based on several subjectively coherent, but temporarily distributed sessions.

5.3.4 Infrequent Revisits

Leaving the area of short-term revisits, this study underlines the importance of proper support for infrequent long-term revisits¹. They occur quite often. Most of them are revisits to modified content.² Below popularity rank 9 of the most frequently visited pages it is common that more than one week has passed between two revisits (see Herder et al. 2006: 286). The median since the last use of a bookmark as found by Abrams is about 100 days (1998: 46). Such infrequent long-term revisits tend to be passed over in many analyses since they are not eye-catching in terms of numbers. Possibly due to this fact they are also undersupported in several history environments. In many statistics they are hidden by the large number of very frequent revisits.

To analyze infrequent revisits in more detail we asked our subjects about such situations in the second interview. They were presented excerpts of their log files that showed situations in which pages were revisited only 2 to 4 times at all, but with several days between the revisits. It was easy for them to identify these situations of infrequent, long-term revisits. In many of them they reported that they actually encountered problems when trying to execute the revisits. One user said he had problems revisiting a physician's personal home page that was hidden in the depth of a hospital's website. He had forgotten the spelling of the person's name and was hardly able to relocate this page. He mentioned that this effort wasted both time and attention and was quite frustrating. Two users reported problems with revisitations to auctions and search results on eBay. They had to reformulate the queries exactly as before in order to find an item again – which was very difficult for them and required several attempts. Another subject mentioned that it was difficult to revisit a researcher's home page hidden within an intricate institutional site. Subjects that reported such situations said they would highly welcome better revisitation support for infrequent long-term revisits.

The mentioned problems can be categorized as follows:

- 1) *Missed bookmarking*. The participants frequently “forgot” to set appropriate bookmarks – due to the fact that they assumed a page would not be needed

¹ An example is “Daedalus’ search for the picture” (situation 4, page 2).

² We made use of the Scone software's ability to create a page's fingerprint. If identical fingerprints were discovered, a revisit was considered a *same content revisit*. About three quarters of short-term revisits were *same content revisits* (73%). In contrast, about three quarters of very long-term revisits were *modified content revisits* (also 73%).

later; or since otherwise too many pages would have to be bookmarked (an argument for continuous logging the complete path); or since they hesitated to clutter their bookmark archives.

- 2) *Forgotten Detail.* Several participants mentioned they frequently forget URIs, paths, or effective Google queries.
- 3) *Problems with the History List.* Some mentioned having considerable trouble using the history list due to missing full text information, misleading titles, missing visual hints, missing search filters, and inappropriate sorting mechanisms. Several did not even know about its existence.
- 4) *Auto-Deleted History-Entries.* Two users complained about items that were deleted from the history list and the auto-completion feature by the web browser for irreproducible reasons.

In the SessionGraphs tool the support of infrequent revisits is supported as soon as the SESSIONGRAPHS-ACTIVE mode is used. Then, all visited pages are recorded and several possibilities exist for long term revisits to a desired target – more than a single other history tool offers for such situations: Textual search (description, title, content) – but also the context of the user’s current task and visual recognition in a session or task visualization. These possibilities could be enriched by additional features such as sorting by name and date. Additionally, with its considerably smaller search space and the possibility to reference items in the deep web the concept provides a big advantage over web wide search engines.

The four problems mentioned above are either not existent, or drastically reduced in the SessionGraphs approach. Missed-bookmarking (in the SESSIONGRAPHS-ACTIVE mode) and the auto-deletion of entries will not occur. The problem of forgotten detail is narrowed by the option of task oriented navigation in the history. The problems with the history list are reduced by the visual presence and the memorizability of the session visualizations. They will be further reduced once improvements such as more meaningful node-titles and others (see page 158) are applied.

5.3.5 Multiple Windows and Tabs

The statistics for all our users showed the mean number of open windows or tabs per T-session was 4.7, with a high standard deviation ($\sigma = 7.1$). The median of open windows per T-session is 2, meaning that in half of all T-sessions at most two windows were used. Usually, the more pages were visited in a T-session, the more windows were opened. For T-sessions with more than ten pages, the median of pages visited per open window was 6.5 – indicating a high usage of multiple windows.

The interviews revealed additional insights about the personal motivation of multiple windows usage. Eighteen of twenty-five participants reported that they regularly use multiple windows or tabs. Only five subjects said they preferred to use just a single window – interestingly those were also the users who stated they typically concentrated on only one task at a time. The majority of tab users reported that they usually had more open tabs than open windows. They also mentioned using tabs mainly for breadth search while using windows for non-related parallel tasks. Summarizing, they stated seven motivations for using multiple windows or tabs that are all supported by the SessionGraphs tool:

- 1) To investigate parallel branches (preferably using tabs). In SessionGraphs supported by the different spokes of a hub and spoke visualization or by different session visualizations.
- 2) To distinguish several tasks that are not related (preferably using windows). A frequently mentioned habit was to maintain distinct browser windows for frequently visited “tools” or “services” such as news, web-mail, translation, search, chats, and forums. SessionGraphs supports this behavior with the tasks concept. This insight, however, motivates to expand the SessionGraphs approach by providing an additional way to easily reuse any frequently used web-based tool in any session, e.g. by providing an additional “tools-bar”.
- 3) To compare several pages quickly (switching between windows using keyboard shortcuts or Windows task bar entries was considered faster than link navigation and backtracking). The SessionGraphs user can choose between using multiple windows, a session visualization, or the combination of both for comparison.
- 4) As a reminder for unfinished tasks that should be continued (in several cases even over night and after several days). This can be done by leaving unfinished tasks and sessions in the “temp” task or by creating a special task to collect such instances.
- 5) To store parts of the path that would be lost after using the back button (i.e. to cope with deficits of current back implementations) – accomplished by session visualizations.
- 6) To cope with unsatisfying slow transfer rates by loading multiple URLs synchronously into different tabs and reading those first that complete first (e.g. in Firefox: indicated by stopped animation of the tab’s icon). The SessionGraphs user can still use this technique. It could still be accelerated by automatically loading an earlier session’s history directly into the browser’s cache as soon as an earlier session is selected.

- 7) To load a page directly into a hidden window or tab in the background while continuing to read or use the currently active window. This technique was very popular in combination with Google searches – the search results list was kept in one tab while specific results were opened and loaded to new tabs in the background for later investigation. This is still possible for Session-Graphs users.

Current tab and window support is still at its beginning. The different browsers provide different means and interaction styles for between window and tab navigation. Firefox, for example, enables between-window-navigation by ALT-TAB, and between-tab-navigation by CTRL-TAB. Many participants did not know about the latter shortcut. While windows are usually quickly accessible via the Windows task bar, tabs are usually not. One participant mentioned to regularly use both Netscape Navigator and Microsoft Internet Explorer concurrently, just to be able to separate different tasks based on the different browser brand icons in the Windows task bar. This indicates the workaround character of the current situation. Several participants felt disturbed by cluttered Windows task bars – one even refrained completely from using multiple windows due to this reason. Many users reported meaningless window and tab titles to be a problem. Others mentioned problems including reduced speed during synchronous loading, cumbersome closing of multiple tabs, difficult inter-tab navigation, and confused window states after loading browser content from external applications¹. Several users were annoyed by websites that automatically open additional browser windows and clutter their preferred style of window usage.

These findings clearly show the need of an improved management of multiple windows and tabs. One suggestion how the SessionGraphs concept could be enriched with multiple window and tab support, is introduced in chapter 6.

5.3.6 Revisits to Groups of Pages

A special role play revisits to groups of pages – where the user intends not only to revisit a single page, but several related pages – either sequentially or in parallel. Such group revisits can be a hint for enduring, but interrupted tasks that are split over several sessions – or for repetitive similar tasks that necessitate several identical resources. As reported earlier (page 43) in almost half of all cases when using a bookmark, the user also wants to revisit other pages in the neighborhood.

¹ Mostly an application will load any browser content into the browser's frontmost window. This window, however, might be used for a completely different task – content that the user may lose during this non-transparent process.

Current web navigation aids provide no specific support for such group revisits.¹ This chapter presents a new algorithm for the detection of group revisits that can be used to improve this situation.

Earlier studies (Catledge & Pitkow 1995; Tauscher 1996a: 87-102) analyzed longest repeated subsequences (LRSs) in the assumption that people would frequently revisit identical sub-sequences of their paths. They used modified versions of the Pattern Detection Module algorithm (PDM, developed by Crow & Smith 1992) for their analyses. Tauscher et al. verified the expected negative relationship between path length and frequency. However, found LRSs were rather short and repeated less than twice on average (Tauscher 1996a). The LRS method turned out to be a non-satisfactory method for identifying group revisits (Tauscher 1996a: 87 & 99f). First, by requiring the path to be exactly the same as before, it is far too restrictive. Already a slight deviation from the path results in overlooking the incident². Such deviations are even more likely to occur when users use multiple windows or tabs for navigation. Second, it is sensitive to unwanted noise like repetitive reload actions, for example. Third, the results show extreme recency – found repetitions occur within a short temporal distance.

Tauscher suggested several improvements and argued: “Until these modifications are made to the PDM algorithm, its value as a predictive/analysis mechanism in the hypertext domain will be minor” (1996a: 100).

The SGUSs-Detection-Algorithm

I developed an alternative approach that detects “Session Groups with identical URI-Subsets” (SGUSs) rather than longest repeated subsequences (LRSs). This algorithm was developed after the longitudinal study showed that look-ahead graphs (based on the identification of identical sub-sequences) were used much less than expected. Thus, this algorithm was not tested with the prototype, but is suggested as a future modification.

The underlying assumption is that people revisit identical sets of pages, but not necessarily in the same sequence as before. If so, then it should be advantageous if history tools offered references to groups of pages, not only to single pages. Thus, it was interesting to find out whether such SGUSs existed between different sessions. The goal of developing this algorithm was an ex post analysis of our data. It could also be used to detect group revisits in real-time during navigation – a possible alternative to the currently used look-ahead technique in the SessionGraphs prototype.

¹ Exception: see page 46, footnote 3.

² As mentioned by Tauscher et al. (1997a: 124): “However, we found that there were many LRSs that have URLs in common but are considered distinct because they do not share exactly the same sequence”.

SGUSs-detection copes with the three disadvantages of earlier LRS algorithms (restrictiveness, liability to noise artifacts, extreme recency), first, by not insisting on a certain sequence, second, by sound preprocessing and by not letting occasional pollutions disturb the detection, and third, by avoiding extreme recency by only considering intersession incidents, not intrasession incidents, and by the possibility of declaring an additional temporal interval that has to exist between any two detected groups.

The SGUSs-detection-algorithm splits the log file into technical sessions and retrieves all session groups with identical URI-subsets. A subset is defined as an unordered subset of the URIs visited within a technical session; the minimal number of sessions that contribute to a SGUS is two. Since the algorithm's implementation is rather straightforward, it will not be presented on pseudo-code level here, but by description of in- and output.

The algorithm is parameterized with

- Δt_{\min} – the smallest accepted temporal distance between any two participating T-sessions. This allows for overcoming the recency problem by enabling the researcher to consider only such sessions with a minimum interval in between – like a day, week, or month. The session timeout remains 25.5 minutes.
- $|host|_{\min}$ – the minimum number of hosts that must occur in the set of shared URIs. This allows retrieving for example only such groups that contain URIs of at least four different servers. The number of matching URIs does not need to be specified, since per run a complete list is received, sorted by number of contributing URIs.

The algorithm returns

- *SGUSs* – the list of retrieved SGUSs, each of which is characterized by
- $sessions_{SGUS_i}$ – the list of T-sessions that participate in the SGUS at index i ;
- $URIs_{SGUS_i}$ – the largest subset of shared URIs between all T-sessions at index i ;
- $duration_{SGUS_i}$ – the temporal distance between the first page visits of the first and last T-session.

Example: If a session A is defined by the sequence of visits (a, b, c, d, e) and the visits in session B are (a, m, n, d, b, o), then exactly one SGUS would be retrieved, presuming that Δt_{\min} and $|host|_{\min}$ are satisfied. The value of *sessions* is (A, B), and *URIs* is {a, b, d}. $URIs_{SGUS_i}$ always contains only the largest subset of shared URIs. The algorithm does not create two additional SGUSs with *URIs* {a, b} and {b, d}. If a third session C is added to the example with visits (a, x, a, b, y, z, a), then two SGUSs would be retrieved:

$SGUS_1: sessions = (A, B), URIs = \{a, b, d\};$

$SGUS_2: sessions = (A, B, C), URIs = \{a, b\}.$

URIs	2		3		4		5		6		7+	
	SGUSs/w	ses	SGUSs/w	ses	SGUSs/w	ses	SGUSs/w	ses	SGUSs/w	ses	SGUSs/w	ses
mean	9.1	5.7	3.6	3.4	2.0	3.0	1.1	2.5	0.8	2.4	1.8	2.2
median	6.3	4.6	1.9	3.0	1.2	2.6	0.6	2.4	0.5	2.3	0.7	2.1
min	0.5	2.6	0.3	2.0	0.0	2.0	0.1	2.0	0.0	2.0	0.0	2.0
max	29.9	20.4	14.0	10.8	8.6	9.3	4.4	4.8	3.4	3.3	16.1	2.6
stdev	9.5	3.7	3.8	1.8	2.3	1.5	1.2	0.6	0.9	0.4	3.3	0.2

Table 5-6: Analysis of revisits to entire groups of pages.

Numbers show Session Groups with identical URI-Subsets (SGUSs) per week over all our participants, normalized by weeks, found by applying the SGUSs-detection-algorithm.

|URIs| = number of shared URIs in a SGUS.

SGUSs/w = SGUSs per week found.

|ses| = mean number of T-sessions that participate in a SGUS.

Quantitative Results

The following results (Table 5-6) were created with $|host|_{\min} = 1$ (no server constraints) and $\Delta t_{\min} = 0$ (no temporal constraints). Results for more constrained settings will be presented below. Since the length of study participation differed highly between our participants the results were normalized to SGUSs per week for better comparability. The normalization to a weekly basis seemed more reasonable than for example normalization per session, since the underlying question was “how often could a user benefit from better group revisitation support”, and the weekly normalization would mirror the individual behaviors much better.

The interesting questions were how many SGUSs would be found and how many T-sessions would participate in each SGUS. For interpretation it is helpful to remember that each subject created on average between 4 and 35 T-sessions per week (mean 11.8).

On average, each participant created about 9 SGUSs per week with 2 shared URIs, and 5.7 T-sessions belonged to each group. The number of SGUSs found decreased with increasing number of shared URIs. A mean of about 3.5 T-sessions per weeks contained 3 shared URIs, about 2 T-sessions per week contained 4 shared URIs. Still, about 1.8 groups of T-sessions per week contained 7 or more shared URIs – mostly with only two contributing T-sessions.

This means that on average each user experienced almost two incidents with group revisits of 7 or more pages per week – a good motivation for improving group revisit support. Some participants created such situations even 16 times per week on average.

LRSs	2#	2AF	3#	3AF	4#	4AF	5#	5AF	6#	6AF	7+ #	7+ AF
mean	9.1	5.7	3.6	3.4	2.0	3.0	1.1	2.5	0.8	2.4	1.8	2.2
median	6.3	4.6	1.9	3.0	1.2	2.6	0.6	2.4	0.5	2.3	0.7	2.1
min	0.5	2.6	0.3	2.0	0.0	2.0	0.1	2.0	0.0	2.0	0.0	2.0
max	29.9	20.4	14.0	10.8	8.6	9.3	4.4	4.8	3.4	3.3	16.1	2.6
stdev	9.5	3.7	3.8	1.8	2.3	1.5	1.2	0.6	0.9	0.4	3.3	0.2

Table 5-7: Analysis of revisits to identical sequences of pages.
 Numbers found by applying my LRS-detection algorithm to our data - used for comparison with Tauscher's results. It shows *weekly LRS occurrences (#)* per user and avg. frequencies (AF) for lengths 2-6 and 7 or beyond.

To compare the SGUS approach with the LRS approach I implemented an LRS-detection algorithm similar to Tauscher's (1996a: 88f) and run both algorithms on our data. The only differences between Tauscher's and my LRS algorithm was that mine reflects session borders for better comparability with the SGUS approach and that it only considers intersession incidents, thereby reducing recency effects. Results are shown in Table 5-7. A comparison to the SGUS-detection algorithm can be found in Figure 5-7.

As expected, eliminating the requirement for sequentiality increased the number of detected incidents. The SGUS algorithm detected about 1.3 to 1.9 times as many group revisitation incidents as the self-implemented LRS algorithm.

Between 24.2% and 91.7% of a user's T-sessions contained SGUSs of length

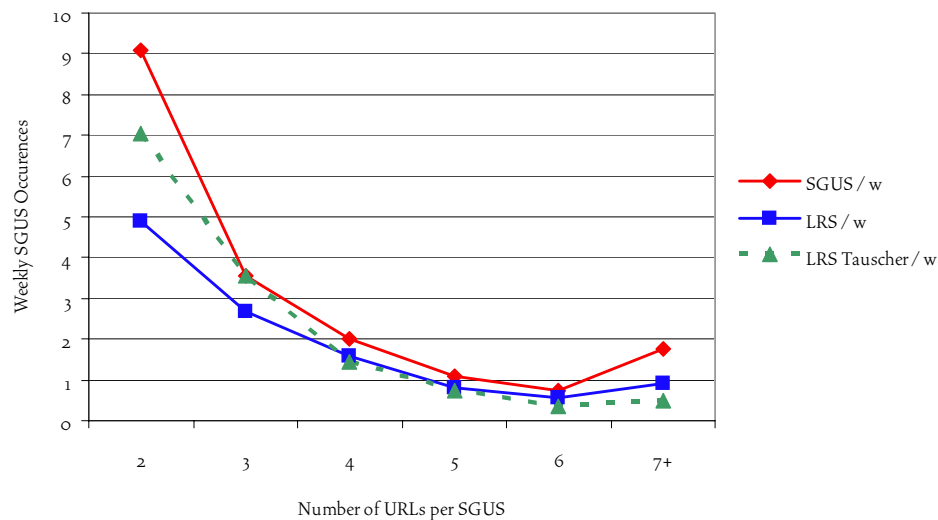


Figure 5-7: Comparing revisits to identical groups versus identical sequences.
 Data found by the implemented SGUS (red) and LRS (blue) detection algorithms. Numbers represent incidents detected per week. Additionally, Tauscher's LRS algorithm is shown (green) with results on their data (limited comparability).

Percentage of Technical Sessions that contain certain SGUSs						
Length of SGUS	2 & above	3 & above	4 & above	5 & above	6 & above	7 & above
mean	64.8%	42.4%	28.9%	20.5%	14.9%	11.2%
median	67.2%	42.4%	26.7%	19.7%	11.7%	6.8%
min	24.2%	11.1%	7.1%	2.5%	0.0%	0.0%
max	91.7%	80.8%	77.3%	70.1%	66.5%	62.3%
stdev	17.3%	16.2%	16.7%	15.1%	14.7%	13.6%

Table 5-8: Analysis of sessions and contained group revisits.
How SGUSs of certain length contributed to technical sessions.

2 or above – averaged over all users, about two thirds (64.8%, see Table 5-8). Still, 28.9% of all T-sessions, on average, contained SGUSs of length 4 or above, and, still, 11.2% contained SGUSs of even 7 or more URLs. Individual values for 7 or more URLs even reached 62.3% of all T-sessions. These numbers show that proper support for long-term group revisitations would be very valuable in many occasions.

To find out how the sessions in SGUSs were distributed over time, the algorithm was initialized with several constraining parameters. The results show that many SGUSs contain group revisits after rather long temporal distances (compare Table 5-9). About 92% of all SGUSs with 2 to 4 shared URIs consist of T-sessions with mutual distances of at least 24 hours. For SGUSs with a higher amount of shared URIs (5-14) this share was still 83%. Even when constraining the mutual temporal distance between participating T-sessions to one week or more, the amount of compatible SGUSs was very high: 76% of all SGUSs with 2 to 4 shared URIs and still about 60% of those with 5 to 14 shared URIs.

SGUSs / week	with 2-4 shared URIs	with 5-14 shared URIs
All	12.44 (100%)	2.58 (100%)
Without “Google”	2.45 (19.7%)	0.85 (32.9%)
> 24h between sessions	11.44 (91.9%)	2.14 (83.2%)
> 24h between sessions, min. 3 servers	1.66 (13.3%)	1.42 (55.2%)
> 1 week between sessions	9.46 (76.0%)	1.53 (59.6%)

Table 5-9: Mean weekly number of SGUSs under certain constraints.

This shows that also such long-term group revisits should be supported appropriately by future history tools. In the SessionGraphs concept group revisitation is inherently coupled to the session-based visualization concept. Once a session is selected it is relatively easy to revisit any related page again – even after very long temporal distances.

The high share of Google visits in SGUSs also recommends improving repeated searches peculiarly. At least one Google visit was found in all contributing sessions of 80% of the SGUSs with 2 to 4 shared URIs and still in about 70% of those with 5 to 14 shared URIs.

The meaningfulness of the identified SGUSs was proved by presenting summaries of found SGUSs to our participants in the interviews. Most SGUSs were verified to consist of meaningful, intentional sessions. Some T-sessions were “polluted” by visits to news or forum pages, which some subjects habitually visit every now and then, simultaneously with other intentional sessions. Many SGUSs consisted mainly of “tool” visits, when participants reused web-based tools to work on similar tasks. Several SGUSs also consisted of “content revisits” where users continued a specific task or re-read specific content. A few examples underline these findings (for details like URLs see appendix 9.6).

Example 1: One participant working in the electronic trading business continuously retrieved data sheets about electronic circuits using the same web-based tools. One SGUS contained 7 T-sessions with 5 shared URIs – split over 5 different servers. The T-sessions were distributed over 84 days. They contained 47 distinct URIs, on average, and the session internal recurrence rate was always about 50%. These grouped “tool revisits” are a clear argument for improved group revisitation support. Using current technology, 5 different bookmarks would have been necessary in order to reuse these tools. Using SessionGraphs, a single session visualization could have been reused.

Example 2: The same subject created one SGUS with 2 T-sessions and 21 shared URIs, about 3 days apart. 86 and 36 distinct URIs had been visited in these T-sessions, in total. The task was similar to that in the first example. This shows that two temporally distant T-sessions may also contain a high number of shared URIs. Appendix 9.6.2 presents five additional examples from other participants.

Summarizing, the SGUS-detection algorithm identified almost twice as many incidents of group revisits compared to earlier LRS algorithms. Our participants produced an average of about 15 T-sessions per active week. Almost half of all T-sessions participate in SGUSs with 3 or more shared URLs. Over 10% of all T-sessions, one to two per week, contain SGUSs with even 7 or more shared URIs. Many SGUSs comprise T-sessions that are temporally very distant and many SGUSs have been identified as personally meaningful for the subjects in the interviews. These are clear arguments for making better group revisitations a prior goal of future history systems.

5.4 Conclusions

The conducted study represents an overdue step with respect to updating existing knowledge about actual user behavior on the web. Major changes compared to earlier studies comprise the lower use of the back button that was only about half as high as before. This was partly caused by the shift to multiple-windows- and multiple-tabs-usage, which now substitutes earlier hub-and-spoke patterns, and by the shift to a more dynamic web including higher usage of form environments that offer their individual back and undo functions. This finding was related with a mean recurrence rate of 49%, relatively low compared to earlier studies. Additionally, we found a very high usage of Google pages, partly used as a means for revisitation. On average over all subjects, more than half of all sessions contained at least one visit to a Google domain. Over all users' visits, a mean of about 17% were visits to a Google domain. When used as a revisitation tool, several problems were mentioned, such as the inability to reproduce a former query, changed result lists, items hidden within too large search result lists, and pages that were not at all covered by Google.

Besides these trends based on the calculated means it should be considered that the intersubjective variances were considerable and that users obviously apply very different styles of using the web. Being adaptable to such different behavior will thus be a central requirement for future history systems.

Several of the found characteristics confirm important aspects of the SessionGraphs approach, others suggest that certain features should be reconsidered.

Amongst the findings that confirmed the main concepts of the SessionGraphs approach was the validation of the session concept, both supported by the subjective comments gained in the interviews and by the measured data. Technically identified sessions contained on average 28 visits to 18 distinct pages – a number that suits the constrained screen size of SessionGraphs session visualizations. T-sessions of very high order were very rare. The highest individual mean was 71 distinct pages per T-session and 97.6% of all T-sessions contained 100 or less URLs (compare page 185). Even these values are still manageable with SessionGraphs session visualizations, as shown in chapter 6.1.1.

With 70.8% of all revisits being either pure intrasession revisits or inter- and intrasession revisits, the cost hierarchy of accessing history items applied in the SessionGraphs approach seems to be well chosen: The majority of all revisits can be executed using the easiest way that SessionGraphs provides – the interaction with a node in the currently session visualization.

The interviews confirmed that the borders of the technically identified sessions mostly matched those that the subjects identified themselves retrospectively in the interviews and several helpful ideas were contributed for further improving the session detection algorithm. Our participants reported to frequently resume ses-

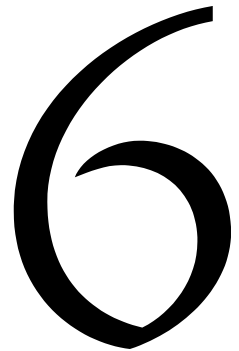
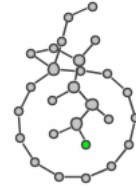
sions after considerable time; some reported that they executed parallel sessions from time to time.

The need of adequate history support for infrequent long-term revisits has been substantiated. As our participants asserted, long-term revisits are often of high subjective importance, but neglected by current history systems. The SessionGraphs tool provides improved support through the explicit representation of tasks that can be used for later revisitation, by means of automatically shown look-ahead graphs, and by additional textual search features.

A further finding that supported the SessionGraphs approach was that long-term group revisits turned out to be actually important for our subjects and that they were often inappropriately supported. The existence of group revisits was validated and it was shown that they can be detected more appropriately by easing the constraints of earlier algorithms for group revisit detection, e.g. LRS. A new algorithm was presented to detect “Session Groups with identical URI-Subsets” (SGUSs), which is able to distinguish about twice as many group revisits. It may be applied in future SessionGraphs versions to improve the look-ahead feature but also in other history tools to offer more appropriate support for long-term multi-page revisits.

Other results of the study, however, showed that the SessionGraphs concept should partially be reconsidered to accommodate the new conditions currently prevalent on the web. These features include the high amount of Google usage, which brings up the necessity for improved integration of search engine use in history systems, not only for first time visits, but also for revisitation purposes. In addition, the increased usage of multiple windows and tabs lead to new forms of short-term “backtracking” by switching back and forth between the different windows and tabs. The current model underlying the SessionGraphs still focusses single-window browser use with short-term revisitations caused by traditional backtracking. Future versions, however, have to consider multi-window and multi-tab navigation in particular.

The necessary modifications, as well as ideas for improvement that emerged during the studies with the SessionGraphs prototypes, will be addressed in the upcoming section.



Suggestions for Further Improvement and Future Work

The lawyer has at his touch the associated opinions and decisions of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest (Bush 1945: 45)

This chapter presents ideas for potential improvements to the SessionGraphs approach – based on the insights gained in the three SessionGraphs studies and in the web usage study. After presenting the five most effective potential improvements, several smaller ideas for beneficial future work are summarized.

6.1 Five Specific Suggestions

The five most important areas for improvements include:

- *modification of the layout algorithm* to create less space consuming visualizations,
- *semi-automatic detection of tasks* to reduce the current organizational burden,
- *improved detection of group revisits* to provide support similar to look-ahead graphs, however, to offer it in more situations by releasing the algorithm's constraints for activating the look-ahead graphs,
- *characteristic representation of search engine usage* in the session views, and,
- *visual support* for the navigation between *multiple windows and tabs*.

The implementation and evaluation of these suggestions exceeds the scope of this thesis; they will be introduced only briefly below. They are, however, considered valuable candidates for future work.

6.1.1 Less Cluttered Views: A Denser Layout

The current session graph layout produces good results (characteristic, individually appearing session graphs) when the visualized sessions contain several revisits. Several subjects of the longitudinal SessionGraphs study asked for a more compressed layout for visualizing more nodes in a smaller screen area, which would also delay the moment of necessary scrolling, especially for situations in which long sections of linear link following occurred without intermittent revisitations. Two suggested using more spiral-like graph layouts for this purpose rather than the linear “zig-zag” shapes that emerge when the current algorithm is used.

The placement of nodes is a result of the *place*-algorithm that defines the insertion point of a new node (see page 122), and the *relax*-algorithm that determines the animated repositioning of all visualized nodes until a balanced layout is reached (see page 118). The improved solution presented in the following is based on a few modifications of the *relax* method, which lead to a considerably improved visualization. The two main driving aspects are, first and foremost, to reduce edge crossings by finding an insertion point that is further away from the bounding rectangle’s center than the currently visited node. And second, to add a radial offset to the straight line intersecting the center and current node. This always identical radial offset causes a spiral-like growth instead of the zig-zag shapes when several new pages are visited consecutively.

Two additional, smaller changes include re-defining the distance between the outer rectangle and the bounding rectangle, not as a static value, but so that it increases dynamically as the bounding rectangle grows, and to determine the final insertion point based on a maximum distance d_{\max} to the current node.

The improved *place*-algorithm for determination of the insertion point of new nodes is described in the following (see Figure 6-1, compare also page 122):

- 1) A bounding rectangle BR of the current session graph and the current node CR are given; the insertion point P for the new node has to be determined. Variables: $d_{\max} = 100$, direction *dir* = counterclockwise.
- 2) Determine the center C of the bounding rectangle BR.
- 3) Determine an outer rectangle OR with the same center C that is horizontally and vertically 20% larger than BR.

- 4) Determine a straight line g_1 defined by C and the *current node* CR .
- 5) Create a straight line g_2 through C by adding angle $\alpha = \frac{\pi}{4}$ to g_1 in direction dir .
- 6) Determine the two intersection points P_1, P_2 of OR and g_2 .
- 7) Choose the intersection point P' (either P_1 or P_2) that is closer to the currently active node CR , or P_1 if both distances are equal.
- 8) If the distance $|CR P| \leq d_{max}$ then $P = P'$, else determine P in distance d_{max} from CR on a straight line g_3 through CR and P' .
- 9) Place the new node at insertion point P .
- 10) Whenever a page is revisited in the current session, change the direction dir to the opposite direction (from counterclockwise to clockwise, or from clockwise to counterclockwise) and then keep it for all subsequent cases.

Two special cases are treated separately, which is kept identical to the earlier version. When the new node is the first node in an SG-session, it is placed in the center of the view. When the new node is the second node in an SG-session, it is placed at a certain predefined distance and angle to the side of the first node. The angle is determined in combination with a random factor so that any two session visualizations will not look the same.

The new algorithm was tested by an a posteriori visualization of real web sessions, whereby the navigation data were collected during our 2005 plain web browser study. Figure 6-2 illustrates the different outcomes obtained when applying

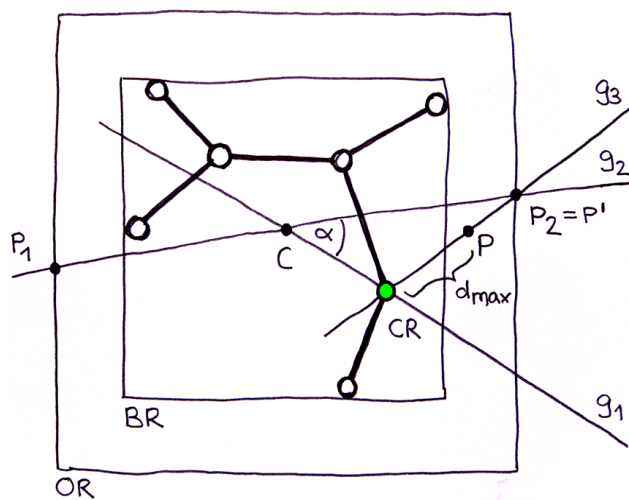


Figure 6-1: Improved node-placement to create a denser, spiral-like layout.

the new versus the old *place* method to exactly the same T-sessions containing the currently problematic areas of sequential link following with only a few revisits: the left visualization is always created with the new algorithm, the right one with the old. As can be seen, the new method tends to create more spiral-like shapes that use less rectangular screen real estate than the old method. In addition, the number of edge crossings can also be reduced in most cases. An important benefit of the new algorithm seems to be that it creates more easily memorizable, different shapes than the earlier one, mainly since the earlier zig-zag shapes often looked very alike. For example, users could be able to remember an item's position more easily using the hours of a clock, e.g. "it is located at four o'clock" (on the surrounding spiral arm), versus "it was about the seventh of the eighteen nodes on the zig-zag line extending from the center".

An additional numerical analysis confirmed the improved use of screen space. The log-data of 100 randomly selected technical sessions of our 2005 web usage study were laid out using both the former as well as the new layout algorithm for test purposes. They were chosen across all the participants with the sole restriction that they contain more than ten nodes, since the problem of disadvantageous screen space usage or zig-zag shaped forms only occurred with sessions around this order or larger.

The results show that the session visualizations, when laid out with the new algorithm, used on average only 66% of the screen space compared to the earlier algorithm ($\sigma = 24\%$, compare Table 6-1). The median was only 62%; the minimum being 26%, the maximum 135%. Especially sessions with very few nodes sometimes needed more space when laid out with the new algorithm. The space saving effect became larger when sessions contain circa 20 nodes or more. These sessions required approximately 61% of the screen space than before (compare also Table 6-1).

Further examples depict the visual character of the session views that the new algorithm creates. Figure 6-3 presents 80 session visualizations laid out using the *new* algorithm (here shown as miniature graphs) based on actual technical sessions that were randomly selected from the 2005 web usage study's log files. All visualizations were automatically generated, untouched by hand. These examples vividly illustrate the unique shapes the new algorithm creates.¹ Concluding, the new algorithm can rightfully be considered an improvement to the earlier solution.

Ideas for future potential enhancements include creating even more characteristic shapes by allowing straight sequences to form from time to time between the

¹ These layouts do not consider that the algorithm only connects two previously unconnected nodes if the recent visit was caused by manual interaction with the *browser interface*, not with the *SessionGraphs interface* (see page 81). In the shown examples, edges were automatically inserted for any simulated step of navigation. Thus, the depicted graphs contain considerably more loops than they would when being created in a real setting.

	11-65 nodes (all)	11-19 nodes	20-39 nodes	40-65 nodes
mean	66%	71%	61%	62%
stdev	24%	23%	23%	26%
median	62%	68%	58%	58%
min	26%	42%	28%	26%
max	135%	135%	104%	107%

Table 6-1: Screen space saving character of the new layout algorithm.

The values represent the share of required screen space when the new algorithm is used compared to the space needed when using the earlier algorithm.

spiral-shaped arcs, further reducing edge crossings and the overlapping of node-titles by a fine-tuned algorithm, as well as using thumbnails instead of circles.

6.1.2 Reduced Overhead: Automated Session & Task Organization

Participants of the longitudinal SessionGraphs study raised the desire for semi-automated support of sessions and tasks in order to reduce their organizational load. They contributed various valuable suggestions.

Sessions

The following situations in the user's navigation behavior could be used to tell the system automatically when to create a new session. Due to the broad inter-subjective variance, the user should be able to adapt these rules, switch them on and off, and potentially create new ones.

The system could start a new session automatically

- whenever the system starts – as already implemented;
- whenever a new window is actively opened by the user and not by the page's source code; in contrast, for many users, opening a link in a new tab should be interpreted as still belonging to the same session;
- whenever a page is opened from an external application such as an email tool or a PDF document;

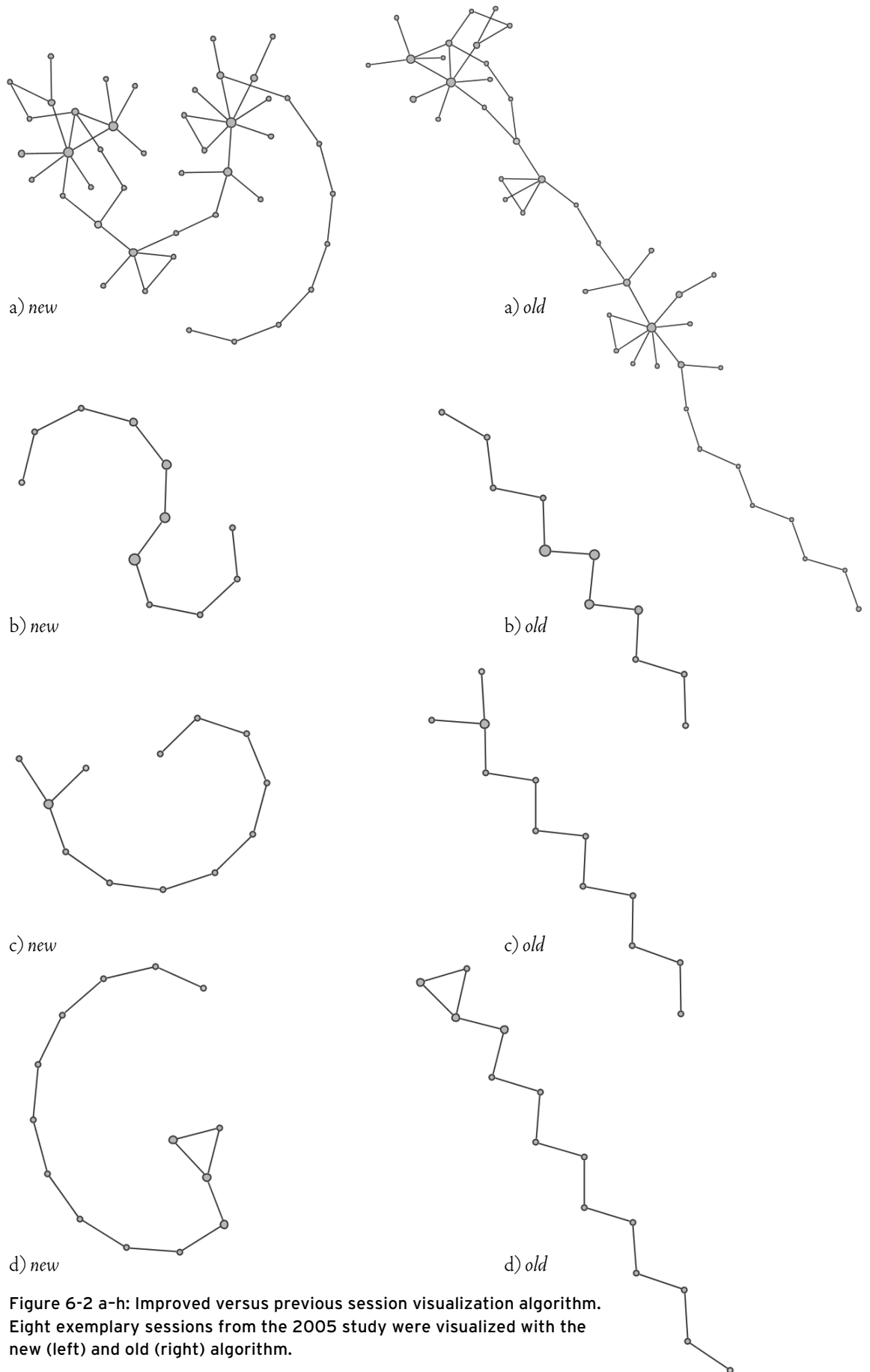


Figure 6-2 a-h: Improved versus previous session visualization algorithm. Eight exemplary sessions from the 2005 study were visualized with the new (left) and old (right) algorithm.

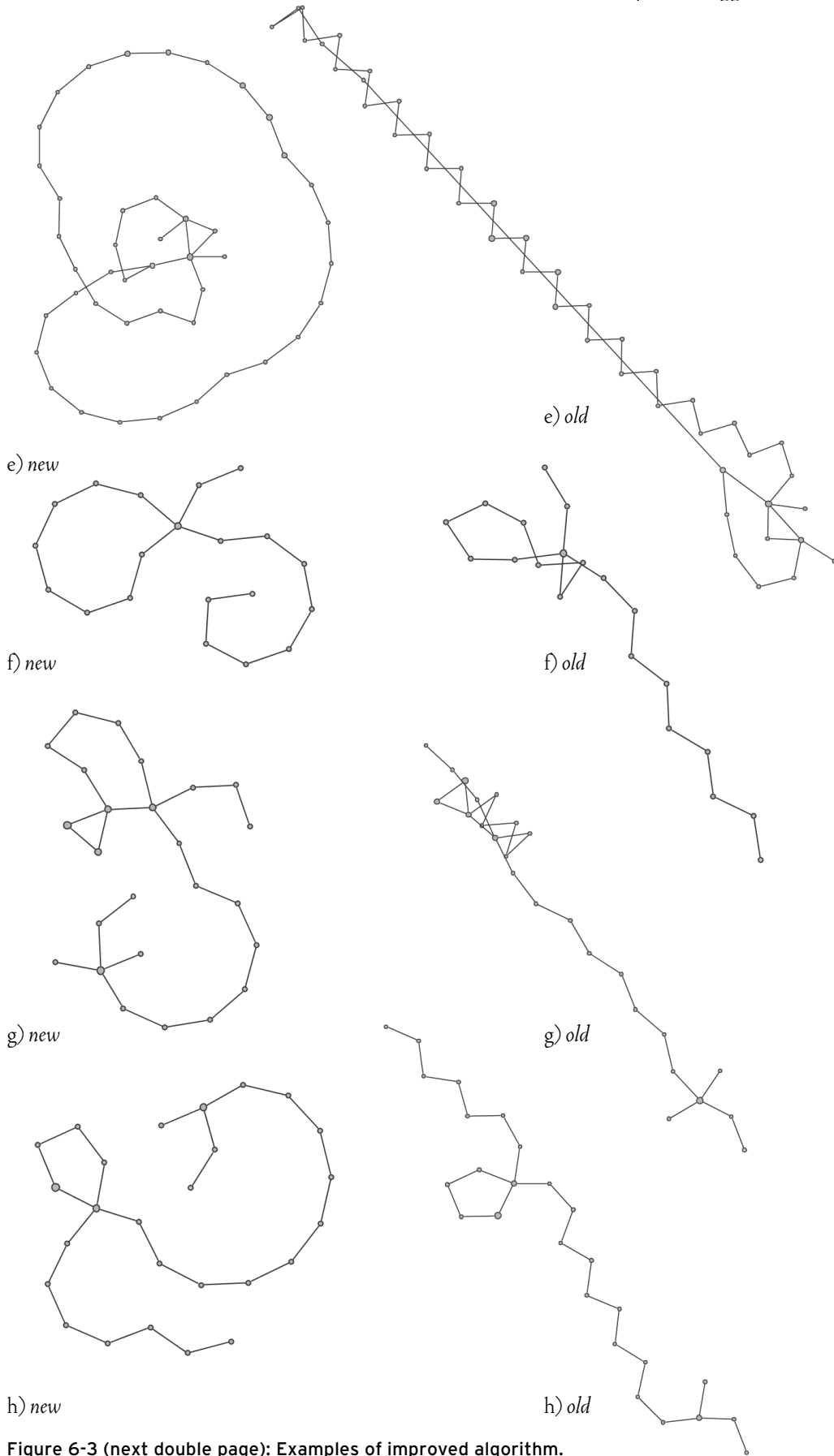
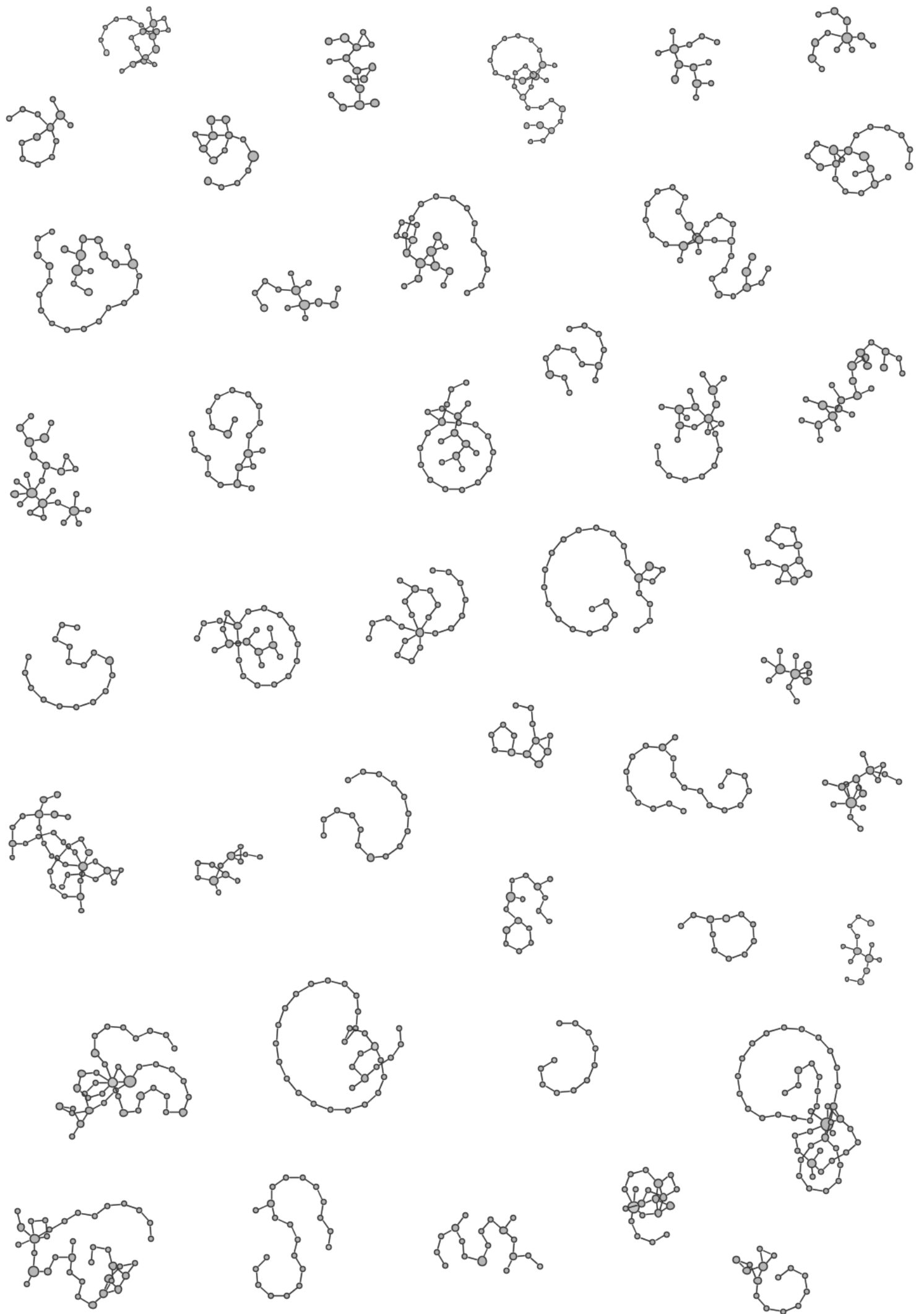
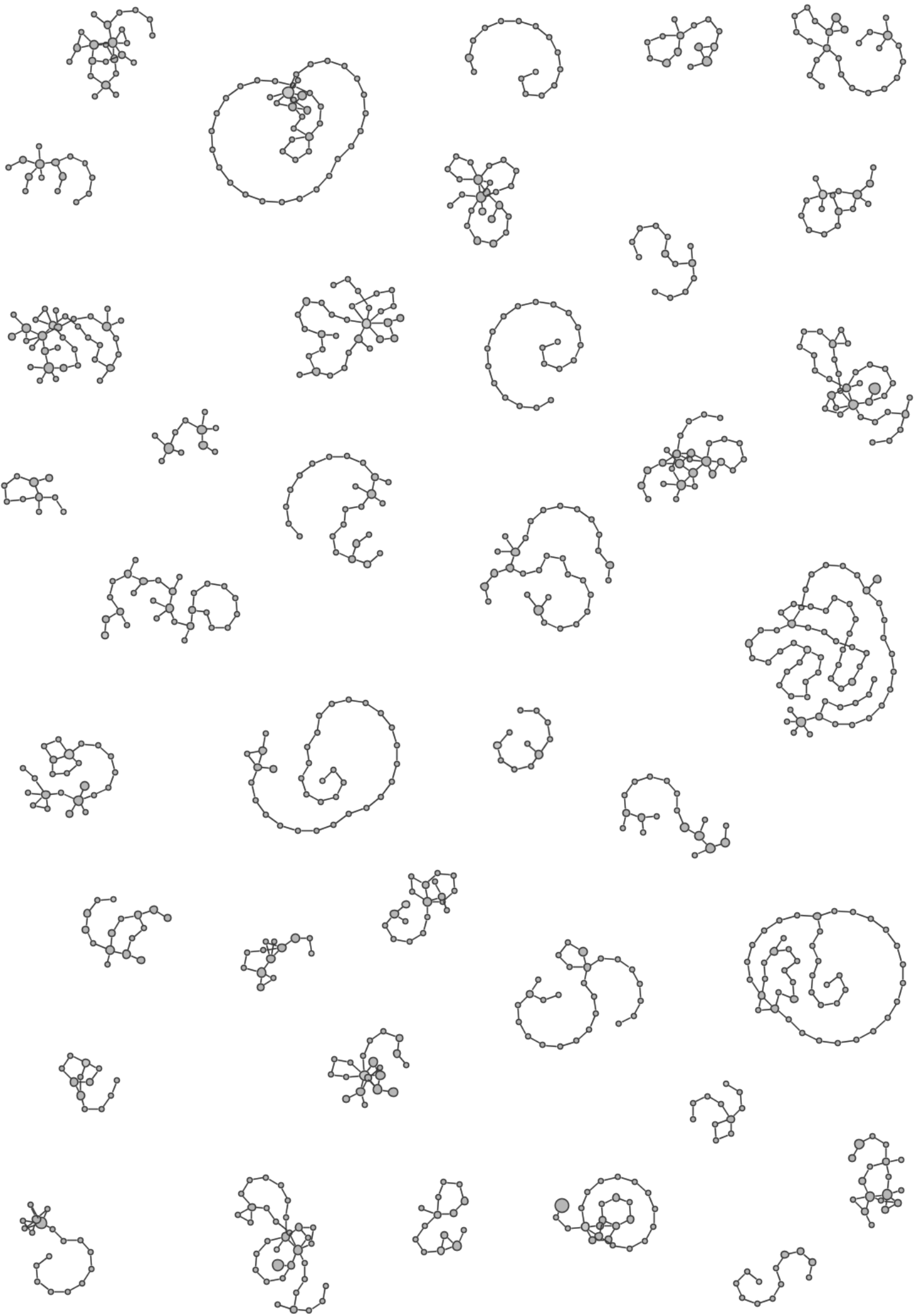


Figure 6-3 (next double page): Examples of improved algorithm. Eighty sessions of the 2005 study visualized with the new algorithm.





- when a page is opened from the browser’s bookmarks or history list. However, the successive use of bookmarks or the history list – if executed within a certain time span – should be interpreted as still belonging to the same session;
- whenever an interval of 25.5 minutes passes without a new visit (compare definition of technical sessions, page 182). A small field could communicate, via a color code, the current state of the system. For example, before the interval expires, a green color indicates that the next visit will be interpreted as still belonging to the current session; a blue color – after the interval has passed – indicates that the next visit starts a new session, opened in a new window; blinking green and blue, finally, indicates that the interval will expire within the next five seconds – to minimize the chance that the user will interact at the moment of state change, resulting in an unwanted behavior. The system should additionally provide a means (such as a key shortcut or a button) to manually toggle the current system’s state independently from the time interval so that the user can start new sessions before the interval expires, but also continue earlier sessions thereafter.

Tasks

One shortcoming of the presented approach is the overhead of manual task and session organization, necessary as the history grows. Bookmarks, however, show a similar shortcoming, and only slightly more than half of the users are willing to spend this additional overhead (cf. page 42).

Recent projects, mainly in the area of Personal Information Management (PIM), have designed and examined new ways to enable a person to utilize large collections of already known items, such as a local file system or personal notes, by creating automatically indexed “flat” structures that are extremely fast and easy to search. Examples include the NoteLens note-taking tool by Benjamin Bederson (2007) and the two PIM tools *Stuff I’ve Seen* (Dumais, Cutrell, Cadiz, Jancke, Sarin & Robbins 2003) and *PHLAT* (Cutrell, Robbins, Dumais & Sarin 2006), the latter largely based on the insights gained during the KFTF project (see page 34).

NoteLens, a speed-optimized, highly focused note-taking tool integrating with Outlook, provides instant access to textual notes, contact information, and emails, and is explicitly designed to support users in their *flow* experience (Bederson 2004, see Figure 6.4). It supports the user in quickly capturing a thought and finding it again later without being considerably distracted from the actual task. Notes can be created during any activity starting with a single keystroke; the first line will automatically be used as the subject. Re-finding is accomplished by typing a word into an entry field, causing all notes that contain the word to be immediately pulled

up, applying a form of dynamic querying (Ahlberg et al. 1992; Ahlberg & Shneiderman 1994). Users can apply filters and create categories to fine-tune their search.

A similar approach is used in PHLAT, the successor of Stuff I've Seen, a Microsoft Research project to support personal search and organization, enabling the user to quickly access any earlier used resource, be it a locally stored document, an email correspondence, a picture, a calendar entry, or a document on the web, which also offers additional features for filtering and querying. As textual search interfaces, these tools are still based on recall, and not on recognition.

Merging these ideas with the SessionGraphs approach could result in a tool that is able to automatically maintain all the web sessions that a user creates without having to manually organize them in a task hierarchy, relieving those users who do not want to spend time and attention on additional organization. Instant, fast textual search could be used to retrieve any previous session and ultimately improve the

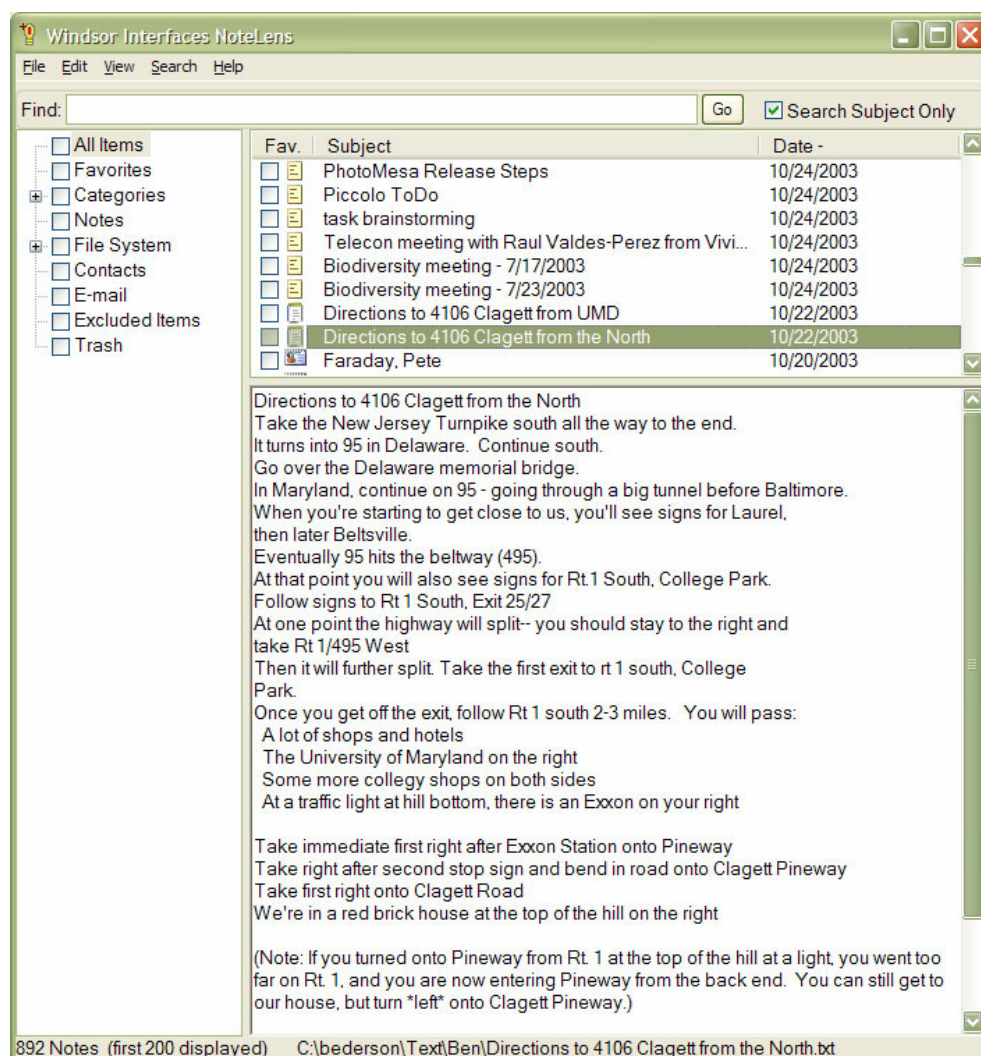


Figure 6-4: The NoteLens Interface by Benjamin Bederson.
 A focused, speed optimized note taking tool integrating with Microsoft Outlook.

experience of *flow*. This could work even better if the full text of all visited documents would be stored, and if the users were able to tag and add meta-information to single documents and whole sessions.

Textual search based on recall could also be used to specify an earlier session (currently accomplished by the task chooser), whereas recognition could be used once a session was selected and visualized. Moreover, the task view could show related sessions that are automatically determined in a background process, instead of showing sessions of the same task, which requires the user to explicitly define and organize tasks.

One promising approach for automatically determining related sessions is outlined by Susan Dumais (Dumais 2007). She uses *contextual information* to create a notion of the user's current activity and to determine further relevant information that should be presented to the user (compare also the concept of contexts presented on page 17). She distinguishes between *user contexts* (short- and long-term interests of the users, their behaviors), *object contexts* (e.g. inter- and intra-object attributes and relationships), as well as *task* and *social contexts* in which information needs arise. Using the principle of *finding without searching* (Cutrell et al. 2006), this contextual information could be used to present potentially relevant earlier sessions. A second way to determine the relevance would be to calculate *session similarities*, for example, by extracting keywords from the single visited pages. One approach based on topic maps is presented by Mase (2005). Visited pages and sites could be compared based on document similarity (see Tiernan, Farnham & Cheng 2003; Baeza-Yates & Ribeiro-Neto 1999; Broder, Glassman, Manasse & Zweig 1997).

A shortcoming of both solutions is that it is not predictable which sessions will be chosen and presented; the user does not know for sure which sessions will be offered based on the current activity. Furthermore, the spatial layout of the task views would probably be less meaningful since the presented items would change often and automatically. A related question is how to prevent the distraction potentially caused by a constant change of the presented contextual session information based on what the user does (compare Cutrell et al. 2006: 64).

While this approach seems very promising for reducing the overhead of manual maintenance, further research is necessary regarding how to apply it satisfactorily.

6.1.3 Improving Look-Ahead Graphs

Look-Ahead Graphs currently support the user in long-term revisits which single out target pages that he habitually revisits by navigating along a remembered path. The longitudinal web study showed that people are likely to revisit groups of pages in temporally distant sessions, however, without necessarily following the exact

sequence as before. The SGUS detection algorithm was introduced to notice such events.

This algorithm was used for analytical purposes in the precedent chapter. It could be integrated into the SessionGraphs tool to replace the current algorithm that activates look-ahead graphs. Instead of detecting same sub-sequences it would detect same sub-groups. Following this approach, the incidents in which look-ahead graphs are activated would roughly double thereby supporting the user in many more long-term revisitations.

6.1.4 Visualizing Searches

The high usage of search engines and the typical constellation of search-related pages (query page, results lists, navigation, and target pages) argue for a special visual treatment of search-related navigation. The user should be able to immediately *perceive* the different types of search-related pages and the structure of the current search. One suggestion is sketched in Figure 6-5. “A” represents the initial query formulation, which should also be viewed as a separate visual item, when the search was originally executed without the use of a distinct web page, but rather by e.g. a built-in toolbar search field. Consequently, each query has its own corresponding visual representation, which, if interactive, could also be used for query reformulation. In the example, the query is represented as an editable text field showing all query parameters. The user could directly interact with this field to add, remove, or

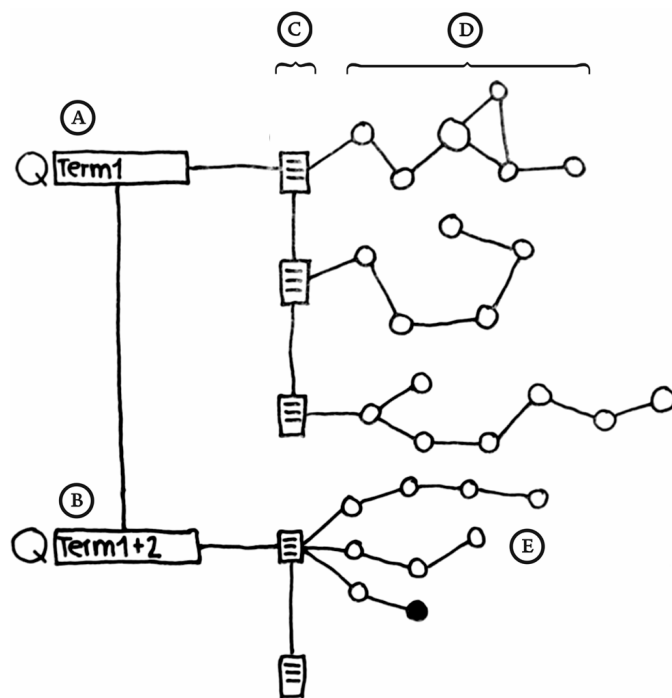


Figure 6-5: Suggestion for an improved search visualization.

modify query parameters, which would result in another visual query node “B”, the submission of the query to the search engine and the subsequent visualization of the search results. Search results pages (“C”) are then aligned on a vertical line to the right of the queries. All of the subsequent search results pages are contained in a rectangular area to the right of the search results node (“D”) to spatially separate different branches. “E” marks an instance where the user explored several results starting from one particular results page.

An additional improvement for long-term revisitations, motivated by the recommendations of participants of the web usage study, is to reproduce the complete results lists exactly as they were encountered during the last visit. This could be achieved by locally storing the lists, as well as by offering the user the choice to review either the original results list or to reuse the search engine again.

6.1.5 Multiple Window and Tab Support

The fourth suggestion for improvement is related to multiple windows and tab support. No current history tool provides appropriate support for navigating between all synchronously opened windows or tabs, or to reconstruct a session properly that was formerly distributed over several windows. The following scheme shows how the SessionGraphs visualizations could provide such support in different situations. Technically, the implementation requires enabling the possibility to exactly trace the user’s navigation between different windows and tabs.

Situation A – “monoview”

Only one browser window is used with no additional tabs. Here, the path should simply be visualized in one corresponding session view as shown in Figure 6-6.

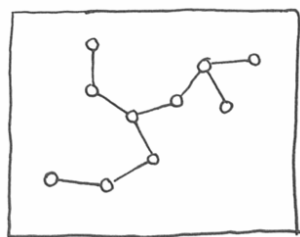


Figure 6-6: Multiple-windows situation “monoview”.

Situation B – “multiview, unrelated”

Several windows or tabs are used for unrelated tasks and sessions. In this case, the histories should be treated as single, separate visual histories per window or tab (see

Figure 6-7), e.g. as separate session view tabs. Once a browser window or tab is brought to the front, the corresponding session view would replace the current one.

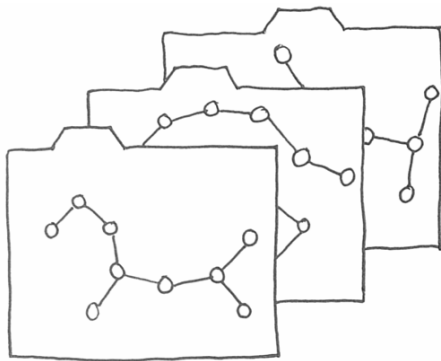


Figure 6-7: Multiple-windows situation “multiview, unrelated”.

Moreover, once a session view tab is chosen by the user, the corresponding browser window should come to the front.

Situation C – “multiview, related”

Whenever the histories of the different windows or tabs belong together semantically from the user’s point of view, it could be worthwhile to treat them as visually coherent entities – or at least to offer this possibility to the user. Current browsers strictly separate the back stacks per window and tab and do not support inter-window-navigation along the path. Considering the high confusion often caused by complex Window bar items, this is a big shortcoming.

A suggestion to improve the SessionGraphs approach is illustrated in Figure 6-8. The upper part depicts the structure with nine open browser windows represented, on a conceptual level, as differently colored rectangles. Arrows indicate which window was opened from another window. The initially opened window represents the root node while the subsequent windows build three additional layers in this example. The exemplary path of visited web pages A to R is distributed over the windows. This node-link diagram already represents a first possible visualization of such a multi-window navigation. In this form, however, it requires an unnecessarily high amount of screen real-estate for visualizing the information.

A space saving approach that shows the same information is represented in the lower part of Figure 6-8; the bottom right shows a possible final visualization, for use in future SessionGraphs versions. The bottom left shows how this spatial arrangement is conceptually derived. The same background color indicates the same window.

The spatial layout is based on the treemap idea invented by Ben Shneiderman (1992). Treemaps depict hierarchies as nested rectangles of certain size. They are well-suited to display hierarchically organized, potentially ordered items of a certain

size in a space-saving way (such as directory tree structures or stock market data). Using a space filling layout strategy, they are better able to deal with the constraints surrounding screen real-estate.

The advantage of this multi-window visualization is that a coherent SG-session spread over multiple windows is represented as a single visual entity. The user can actually see in one overview what already belongs together semantically, which so far was not easily perceivable. Then, he can immediately navigate to whatever node in any window he or she likes by interacting with this view.

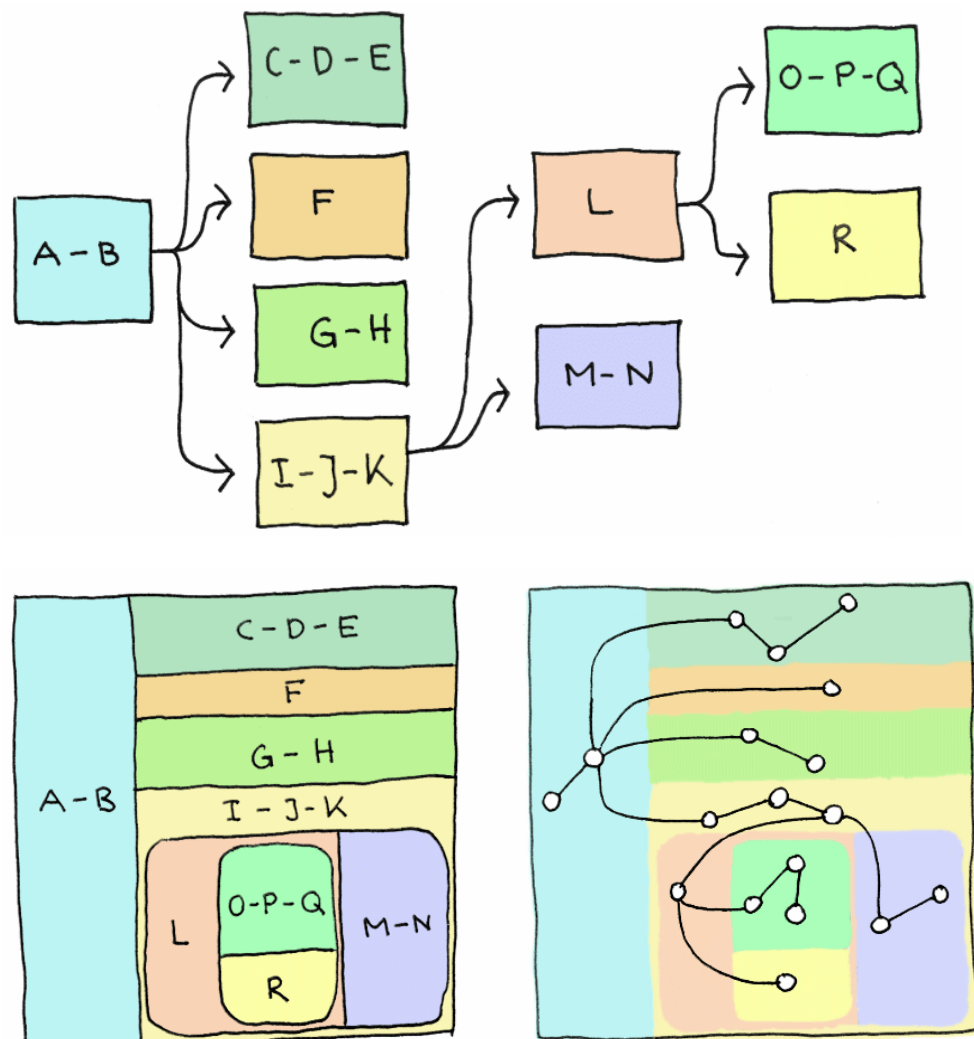


Figure 6-8: Multiple-windows situation "multiview, related".

Top: How the navigation path (A to R) is distributed over the hierarchy of windows or tabs. Bottom left: How this hierarchy could be visualized in a treemap (slice-and-dice layout). This view depicts which visited page will be mapped to which area in the treemap. Bottom right: Same treemap layout filled with the actual session graph - this could be the final form of a multiview-related visualization.

In the suggested visualization, one treemap area comprises the navigation related to one specific window. The example uses the slice-and-dice treemap algorithm for spatial layout (*ibid.*). The paths related to the different windows and tabs are visualized in a predefined rectangle. The rectangles' nesting represents the different layers of the windows hierarchy.

Several other treemap algorithms could be used as well. They differ in metrics, which cannot all be optimally satisfied at the same time (Bederson, Shneiderman & Wattenberg 2002): desirable qualities usually include a low aspect ratio of the created rectangles' sides, a low distance change when data are updated and a high readability – enabling the reader to follow the provided order easily. (*ibid.*). The slice-and-dice algorithm used in the sketch is the earliest algorithm and incidentally provides the highest readability. However, it usually produces undesirable high aspect ratios.

Not recommendable for SessionGraphs are cluster or squarified treemaps, even if they optimize the aspect ratio, since they can cause dramatic discontinuous changes in the layouts when the values attached to each area change over time. The user should be able to locate a window's representation in the view at approximately the same position, even if the window's part of the session graph has changed.

Better alternatives might be strip treemap algorithms and pivot-by-split-size treemap algorithms (Bederson et al. 2002), which improve the slice-and-dice method in terms of aspect ratios and still maintain good readability. The applicable algorithms also depend on whether a window's session graph should be treated as a fixed size element, first laid out and then inserted into a given treemap area, or whether the dynamic layout of the session graph should interfere with the spatial arrangement of the treemap. In the first case quantum treemaps (*ibid.*) would be preferable, while in the latter, all previously mentioned algorithms could be used.

For easier recognition, windows and tabs could be color-coded both in the history visualization and in the web browser (e.g. by border and fill color). Windows could also be distinguished from tabs in the history visualization, e.g. by different border styles.

Using this approach, users can immediately switch back and forth between all open windows and their nodes. They can even reopen a window that was closed before – which would be specially marked in the history visualization, e.g. by different background shading. Presuming a reliable technical coupling of visualization and browser, the user could even drag a node in the visualization from one area to another – thereby modifying the histories of the related browser windows. He could move a representation of one tab into another window's area, thereby moving the tab from one browser window to another. In the short-term, the user could back-track seamlessly over multiple windows' borders. In the long-term, complete window constellations including the single windows' histories could be easily restored.

While this idea presents a first step for treating multiple windows and tabs, several questions still remain. For example, how many items or windows could such a visualization sustain? Are users able to interpret and use such a visualization easily? Answering these questions seems to be valuable and exciting future work.

6.2 Additional Recommendations for Future Work

Beyond these five suggestions for further improvement, various additional features emerged as valuable recommendations for future work. They can be assigned under four different sub-headings: re-implementation, further evaluation, modifications of the existing approach, and alternative visualizations.

6.2.1 Re-Implementation

The SessionGraphs concept was implemented as a research prototype to demonstrate and evaluate the underlying ideas. Based on the lessons learned, a new implementation should simplify future explorations and evaluations. The following two modifications should lead to increased flexibility and extensibility.

First, future implementations based on Java or C# could consider Piccolo.Java or Piccolo.NET as a substitute for Jazz (see Bederson 2006). Piccolo was developed by the Jazz developers with the main goal of making programming even easier. It is based on a monolithic approach that would significantly reduce the number of objects the programmer has to deal with. Piccolo is equally as fast as Jazz with the added bonus that it uses less memory than Jazz. The benefits of easier implementation and the use of an up-to-date technology argue for switching to Piccolo.

Second, the SessionGraphs application should be integrated more closely into the assisted web browser. The chosen proxy solution supported the basic needs well. However, window and tab specific visualizations as well as the appropriate reaction to encountered frame pages or special web-based services require a closer coupling between visualization and browser. An additional benefit of a closer integration would be the possibility to show the SessionGraphs GUI in the browser sidebar, preventing the user from having to switch back and forth, dealing with an extra application window. A promising technological basis is provided by the XML User Interface Language, XUL, developed by the Mozilla project (Mozilla__Project 2003). It provides access to important browser events and an environment for rapid prototyping. Approaches such as those designed in the MIT's SIMILE project (Huynh 2004a, 2004b) may also enable the integration of Java GUIs in combination with XUL.

6.2.2 Further Evaluation

The two controlled experiments and the longitudinal study only evaluated specific aspects of the SessionGraphs approach. Future evaluations should compare the approach not only to a standard web browser, but to other alternative visual approaches, such as the PadPrints concept, the Nestor approach, or the GroupLab system (chapter 2.3.2). Upcoming controlled experiments should shed light on the users' ability to remember and recognize entire session visualizations, but also single meaningful nodes within them. A re-implementation that overcomes the current technical obstacles could enable a longitudinal study of the approach with more participants and less distraction based on technical incidents. As a result, the single contributions of the different functionalities could also be estimated more appropriately (full-text search, look-ahead graphs, animation, etc.).

6.2.3 Modifications of the Existing Approach

The SessionGraphs concept was only presented in one of several possible forms and it would be interesting to study variations of its basic ingredients. Some possibilities are presented below, covering the following topics: improved scalability, improved memorability, the replaying of paths, improved functionality, and the provision of site-graphs.

Improving scalability. The modification to a more spiral layout would be a first improvement to the current scalability of session graphs. Alternatives could be evaluated that allow creating even larger sessions without the drawback of early scrolling, e.g. such that all visited nodes are not visualized or they collapse several nodes to one visual entity. A heuristic could be used, for example, to detect pure navigational, intermediary pages which could then be omitted or collapsed in the view. In addition, techniques such as semantic zooming (Bederson, Clamage, Czerwinski & Robertson 2004: 97), zooming (Bederson, Hollan, Perlin, Meyer, Bacon & Furnas 1996; Combs & Bederson 1999; Bederson et al. 2000), fisheye views (Furnas 1986), and hyperbolic visualization (as applied in Footprints, page 58) may be explored for their suitability in creating more scalable session visualizations.

Improved memorability. Different techniques could be explored for their ability to increase the memorability, first, of whole session visualizations, and, second, of the position of single meaningful nodes within a view. They could utilize human mnemonic abilities, as for example discussed by Yates (2001), and make use of the insights of Darken and Sibert (1996b) about virtual worlds, who found that directional landmarks, gridlines, paths, boundaries, and maps promote the acquisition of a mental map of the space. For instance, more characteristic visual backgrounds could be used, such as the abstract colored and shaded panes in the Data Mountain project (page 58), geographical maps, subway maps (which may motivate a new layout algorithm), or pictures taken from the user's photo collection.

Another promising approach to support the recognition of special nodes in spatially changing graph visualizations is presented by Skopik & Gutwin (2005). They examined graphs that were distorted by a fisheye view technique and found that their subjects improved their ability to re-identify nodes after the layout was distorted when they applied a history technique they call *visit wear*, a modification of *read wear* (Hill, Hollan, Wroblewski & McCandless 1992). The idea is to visually mark a few nodes with which the user interacted recently. These visual marks slowly disappear in an adequate time-frame, so that at any given time only a few nodes in a graph are specially marked. Currently, the SessionGraphs approach only marks the currently visited node. Extending visual marks to the last few visited nodes could help the user to re-identify also recently visited nodes even when the layout is changed significantly by animated layout. SessionGraph nodes could, for example, be enriched by visit wear in form of border highlights of decreasing width, transparent overlays of increasing transparency level, or adjacent marks of decreasing size (Skopik & Gutwin 2005).

Re-playing paths. An interesting supplement would be to enable the user to automatically re-play his or her paths, a concept which is as old as Memex that provided explicit trails. Bush gives an example: "... his trails do not fade. Several years later, his talk with a friend turns to the queer ways in which a people resist innovations, even of vital interest. He has an example, in the fact that the outranged Europeans still failed to adopt the Turkish bow. In fact he has a trail on it. A touch brings up the code book. Tapping a few keys projects the head of the trail. A lever runs through it at will, stopping at interesting items, going off on side excursions. It is an interesting trail, pertinent to the discussion" (Bush 1945: 45). Bush further elaborates on the interaction with paths: "On deflecting one of these levers to the right he runs through the book before him, each page in turn being projected at a speed which just allows a recognizing glance at each. If he deflects it further to the right, he steps through the book 10 pages at a time; still further at 100 pages at a time" (Bush 1945: 44).

Similar behavior could be easily integrated into the SessionGraphs tool, so that the user could replay interesting paths or specially designed *guided tours* at any desired speed. This may be useful for quickly finding a desired page again, for regaining orientation after a considerable time, but also to share discovered information with others. According to (Nielsen 2005) "Guided tours let you collect a series of pages and subsites and combine them with additional material into a new structure that you can communicate to others. This is great for e-learning applications, but also has more pragmatic uses. For example, you might research a business purchase and send your boss a guided tour with the pros and cons of different options". The current SessionGraphs design concept would only need a very small step in order to support guided tours. The application of paths and tours has been examined in numerous projects of different direction. Interesting new approaches may be

created by the combination of these projects' main ideas and the fundamental concepts of SessionGraphs. Amongst them are TextNet (separate storable paths – Trigg & Weiser 1986), NoteCards (an early form of 'guided tours' through hypertext – Trigg 1988), the Scripted Documents system (conditional, programmable paths and path playback – Zellweger 1989), the Walden's Path project (guided tours in educational systems – Shipman, Furuta, Brenner, Chung & Hsieh 1998; Furuta, Shipman, Marshall, Brenner & Hsieh; Dave, Karadkar, Furuta, Francisco-Revilla, Shipman, Dash & Dalal 2003), and the Webwise system (applying the metro map metaphor to paths on the web – Sandvad, Grønbaek, Sloth & Knudsen 2001).

Improving Functionality. An additional supportive feature might be the provision of locally stored versions of earlier page content, as well as letting the user choose between the current version of a page, as available on the web, and a locally stored version that was seen before (as e.g. provided by the Scrapbook extension for Firefox¹). This could also be combined with the introduction of personal annotations. As a result of their study, Fu et al. (2005) remark "With more and more reading being done online, most participants are enthusiastic about the possibility of making annotations on web pages". About collaborative and personal annotations on the web see also (Zheng, Booth & McGrenere 2006; Marshall & Brush 2004; Marshall 1998).

Providing author-defined site-graphs. A next valuable step of user support at a website level would be to give the authors of a site the ability to create predefined site-graphs, i.e. visual guided tours, which would provide an additional navigational aid especially for highly interlinked, complex sites, as Zellweger points out (1989: 1): "In most current hypertext systems, readers may fail to understand the material presented because they view it in the wrong order, or they may simply comprehend it less well". Site-graphs could appear in a similar way to session visualizations and could be modifiable by the users of the site to meet their personal needs. This would also give them individual navigation support during upcoming visits to the site.

6.2.4 Extending the Vision

Although the World Wide Web is still in its teens, it already looks back on a history of intense growth and usage. It is extremely demanding to develop history approaches that keep up with this rapid development, and that are able to assist the user during these current and upcoming changes, and that are additionally able to risk leaving present, more limited ways of thinking.

Certainly, nobody can tell for sure what the web will look like in 30 years. However, there are some trends that seem to have a good chance for continuation. A fundamental assumption is that the amount of information available will continue

¹ <https://addons.mozilla.org/firefox/427/>

to grow drastically, since more sources will be exploited, more people will be contributing, and more data will be generated. If so, then history tools will be even more important in the future. They have the huge advantage that they only have to deal with the limited amount of information that an individual person already dealt with in the past. This amount will always be many dimensions smaller than the available information, and it most likely will not grow as fast. The challenge then, is to develop systems that give the users the quickest, easiest, and most satisfying access to their individual history.

The following three trends of web development show particularly high potential: web-based information will increasingly be available from anywhere and by any device, it will increasingly be based on sharing and collaboration, and it will be easily able to bring those things together that belong together.

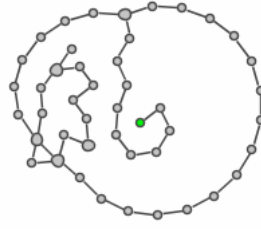
From anywhere, by any device: the ubiquitous web: The term ubiquitous web refers to the ongoing venture of extending the web towards a pervasive platform of applications which provide people with access whenever and wherever they want, using any device of their choice (see W3C 2006; Youn, Kim & Morikawa 2006). This encompasses the devices of the mobile web, but also includes customer electronics, television, automobile technology, etc., and embraces accessibility issues positively affecting people with disabilities. History systems, as well as other applications, will have to be able to adapt to the user's needs, the device capabilities, and the environmental conditions. For example, users could be able to listen to an important paragraph of a document they read three days ago on the web, now using their mobile phone; or, users could tell their hi-fi system to play the song, the title of which a friend sent them a week ago in an SMS. The challenge that history developers are faced with is to create appropriate standards for facilitating the exchange of history information, and to develop appropriate ways of displaying this history information on the various devices. For instance, the presented SessionGraphs visualizations are currently used to display a session on a high resolution monitor, however, they could be rendered very differently on a PDA.

Collaboration and sharing information: Web 2.0: The Web 2.0 trend (see footnote, page 12) that includes collaborative and information-sharing developments, such as social networking platforms (like Orkut, MySpace, Xing), wiki webs (like Wikipedia), blogs (mainly textual ones, but also photoblogs, vlogs, and podcasts), and approaches of folksonomic tagging (like Flickr, and Del.icio.us) opens new chances and challenges for history systems. Interesting questions crop up including, how can history concepts which surpass the idea of singular references (such as the SessionGraphs sessions and tasks concepts) be tailored in order to serve the exchange between several users; what happens when users not only blog their latest ideas, pictures, and comments, but their whole history; and can this then be used for mutual learning and support on a global scale?

Bringing together what belongs together: Web 3.0. Tim Berners-Lee vividly outlines his vision of the *semantic web* (Berners-Lee, Hendler & Lassila 2001; Berners-Lee 2007; Pellegrini & Blumauer 2006), which is also related to the term *Web 3.0*: “Imagine coming home in the evening, entering the living-room, and your mobile phone automatically switches on the light next to your favorite chair. The plasma display on the wall shows the pictures of the great sunset you took on your way home, and the digital video recorder autonomously records the movie that you marked in your office while reading the TV program on your laptop” (Berners-Lee 2006a, translated by author).

Technically, the semantic web aims at creating a “machine-understandable” environment, mainly based on the Resource Description Framework (RDF), a basic model for publishing machine-readable data, and ontologies (OWL – the Web Ontology Language) to characterize the data and their relations more precisely. The notion of a *history* could be largely expanded using the capabilities of the semantic web, since it is no longer constrained to visited web pages, but may comprise any kind of data the user created or dealt with in the past. The computer could “know” about all the technical records of his or her past activities, including calendar entries, pictures, conversations, bank account data, and documents created during a project. Once the issues of censorship and privacy are appropriately addressed, it offers promising possibilities. For example, the system could easily create a presentation of a person’s last trip, including all the pictures taken, personal notes, an online map showing the path taken, and an art-history guide discussing the visited sights. It could assist a person who visits a certain client once per month in New York City by booking the tickets for the next flight, providing access to all related documents and contact information, checking whether the last meeting was already paid, and informing the person that a close friend who was not seen for a long time will also be in New York on that date.

Extending the possibilities related to history usage in the aforementioned ways are exciting, valuable endeavors. Designing the history’s future is a responsible and challenging, but also extremely delightful task that should be able to deliver solutions that could make our lives easier and even more enjoyable.



7

Conclusion and Contributions

*They say that time changes things,
but you actually have to change them yourself.
(Andy Warhol)*

Based on a thorough analysis and evaluation of existing research, current browser history support and alternative web history projects, a new approach was developed that supports the user in all identified situations requiring proper history support, both in the short- and the long-term, and both for single and multiple pages. It applies the benefits of information visualization to the area of web history support by replacing cognitive processes with perceptual ones, thereby freeing users to concentrate on their actual activities. Its constitutional elements are graph-based visualizations of the users' paths that provide references to those web pages that were visited during a single session of web-based activity. These session visualizations can be organized according to superior tasks in a hierarchical way similar to current bookmarks. They can be reused and resumed whenever needed. The SessionGraphs history approach provides one integrated GUI that supports all the different situations of revisitation, using only a single, consistent visual interface.

This approach was conceptually designed, technically implemented, and empirically evaluated with a small number of users. Two controlled experiments, which analyzed revisits within the timeframe of one hour and one to six days, showed that

certain revisitation tasks could be accomplished significantly faster using the SessionGraphs visualizations versus any of the history tools provided by a standard web browser. On average, these revisits also involved significantly fewer visits to pure intermediary navigation pages. Moreover, the subjective user satisfaction, assessed by the application of a QUIS-like questionnaire was significantly higher with the SessionGraphs tool. These results were promising, yet necessarily-based on a controlled setting. Consequently, they encouraged the implementation of additional research.

A longitudinal study was conducted to evaluate the tool's usage under more naturalistic conditions. Five participants used the prototype over several weeks for their actual day-to-day activities. In the first place, they showed that the participants were able to apply the concepts provided by the approach in their regular work. In subsequent interviews, they mentioned the added value of visualization over pure textual lists. Secondly, the study also revealed several areas requiring further improvement. Examples include the organizational overhead created by manual task organization and the automatic spatial layout of the session graphs that sometimes resulted in complex, rather cluttered visualizations or linear zig-zag shapes that wasted valuable screen space. On a technological level, a closer integration of the GUI into the browser would be desirable, e.g. to control the states of multiple windows and the representation of frame pages. In any case, the study was a valuable source of precise suggestions for improvement. For instance, users gave hints about how to automate task management and how to improve the graphs' layout by creating denser, more spiral-like visualizations. The identified shortcomings were then addressed and possible solutions are presented in chapter 6.

The study also revealed, however, that the empirical knowledge on the basis of which the approach was designed, urgently required an update. The web had drastically changed since the studies that originally produced this knowledge – in terms of the content it provided, the tools that supported its use, and the actual methods of usage. In order to adapt and improve the suggested history approach to meet the current needs, a more thorough, recent understanding of the present behavior on the web was required.

An extensive client-side clickstream study was conducted in cooperation with three colleagues to update the findings of the three major studies previously conducted in the field. It revealed striking changes in web usage and demonstrated the necessity of improved history support both in the short- and the long-term. Specific insights include the increasing replacement of short-term backtracking by the navigation between multiple tabs or windows and the high use of web-wide search engines, also for long-term revisitation. An additional finding was that infrequent, long-term revisits are often poorly supported in spite of their high subjective importance. The same was found for long-term revisits to entire groups of pages.

Several suggestions for improving the SessionGraphs approach, based both on the insights gained in the SessionGraph studies and in the click-stream web usage study, are presented in chapter 6. They comprise an improved version of the SessionGraphs layout algorithm, evaluated with actual web-log data gathered in the click-stream study, a suggestion to visually support the use of search-engines and a proposal for how to deal with the navigation in multiple windows.

In conclusion, this dissertation delivers a promising approach for improved visual revisitation support and a thorough update of insights into actual revisitation behavior on the web. The continuation of this work, as proposed in the future work section, seems to be an exciting and worth-while undertaking.

In an environment of lacking history support and outdated insights on actual web usage, this thesis contributes to the development of the next generation of visual web history support and to an improved understanding about the role of revisitation in the current web and what users actually do.

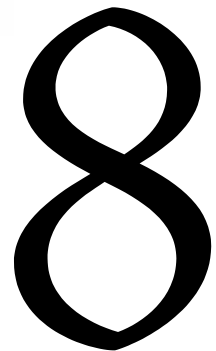
Specific contributions include:

- A *taxonomy of revisit types* and the identification of *seven critical situations of revisitation* to communicate more clearly about the topic of revisitation on the web (chapters 2.2.4 and 2.3.3).
- The SessionGraphs concept including the characterization of web-based sessions (chapter 2.2.3.3) and their *graph-based node-link visualization* (chapter 3.2.2.1), a hierarchical organization scheme based on the user's *tasks* (chapter 2.2.3.1), and the *fluid surface metaphor* for the dynamic visualization of the history (page 78).
- The concept of an *integrative GUI* that supports all seven critical situations of revisitation on the basis of a deliberate cost hierarchy (chapter 3.2.3). This solution addresses the current drawback of an unmanageable multitude of highly different, specialized, mainly text-based history tools, each of which has been shown to contain specific, often severe shortcomings (chapter 2.3).
- Features like *look-ahead graphs* to accomplish same-path revisits, *similar-task revisits* to revisit entire groups of pages, and *group-exchange revisits* to communicate web histories in a workgroup setting (chapter 3.2.4.1).
- The prototypical implementation of SessionGraphs, based on Java, the Scone proxy framework and the Jazz Graphics toolkit (chapter 3.3.2). It demon-

strates one possibility for technically setting up an extensible web history prototype for study purposes.

- The *evaluation of the SessionGraphs concept* under controlled and more naturalistic circumstances (chapter 4). Two *controlled experiments* showed that the approach is a significant improvement to an unassisted traditional web browser (chapter 4.2). A *longitudinal study*, under more naturalistic conditions, shed light on the actual usage of the tool when applied for common day-to-day activities (chapter 4.3). This study also revealed several areas for further improvements. Participants gave valuable feedback and suggestions.
- An *empirical study of the current revisitation behavior* apparent when using unassisted browsers (chapter 5). This collaborative study revealed new insights in how people currently use the web. Revisitation-related insights are published in this thesis. They comprise the low use of the back button, the high use of multiple windows and tabs, as well as the high amount of Google searches. This study also showed the necessity for improved support of infrequent revisits and revisits to groups of pages in the long-term. Results were published in several papers, leading to the receipt of two best paper awards (Obendorf et al. 2007; Weinreich et al. 2006c; Weinreich et al. 2006b; Herder et al. 2006).
- Several suggestions for *potential future improvements* to the SessionGraphs approach in order to meet the requirements identified in the different studies (chapter 6).

The work accomplished in this dissertation improves the ability to clearly communicate about web history-related issues. It presents a consistent, new visual approach for assisting a web user in accessing and utilizing the personal history in appropriate ways. The actual use of the web and revisitation-related issues have been re-studied and updated. Subjective feedback about the strengths and weaknesses of the presented approach has been collected and suggestions have been presented for future development. The sum of these contributions forms a thorough basis from which point future research can continue to improve the web history experience to serve the users' needs.



Used Literature

- Abrams, David (1997). Human Factors of Personal Web Information Spaces. M.Sc. Thesis, Department of Computer Science, University of Toronto, Toronto, Canada. <http://www.dabrams.com/bookmark/> (May 2007)
- Abrams, David, Ronald Baecker & Mark H. Chignell (1998). Information Archiving With Bookmarks: Personal Web Space Construction and Organization. In *Proceedings of Human Factors in Computing Systems (CHI 98)*: ACM Press, pages 41-48.
- Adler, Steven, Jan Heise, Matthias Mayer & Peter Schefe (1998a). CVP: Recognizable and hierarchical 3D Overviews of the WWW by Persistence Directory Placement. In *Proceedings of Computer Graphics International*, Hannover, Germany, pages 404-410.
- Adler, Steven, Jan Heise, Matthias Mayer & Peter Schefe (1998b). CVP: Spatial Representations of WWW-Structures to Enhance Navigation, Presentation and Communication. In *Proceedings of World Automation Congress (WAC)*, Vol. 8, *Soft Computing, Multimedia and Image Processing*, Anchorage, Alaska: TSI Press, pages 379-384.
- Adtech (2007). Browser-distribution by country (Europe) for February 2007. <http://www.adtech.info/en/pr-07-7.html> (May 2007)
- Aguilar, Francis J. (1967). *Scanning the Business Environment*. New York: The Macmillan Company.
- Ahlberg, Christopher & Ben Shneiderman (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proceedings of Human Factors in Computing Systems (CHI 94)*, pages 313-317.
- Ahlberg, Christopher, Christopher Williamson & Ben Shneiderman (1992). Dynamic Queries for Information Exploration: An Implementation and Evaluation. In *Pro-*

- ceedings of Human Factors in Computing Systems (CHI 92)*: ACM Press, pages 619-626.
- Allen, Gary L. (1999). Spatial abilities, cognitive maps, and wayfinding: Bases for individual differences in spatial cognition and behavior. In *Wayfinding behavior: Cognitive mapping and other spatial processes*, edited by R. G. Golledge. Baltimore, MD: Johns Hopkins Press, pages 46-80.
- Altheim, Murray, Frank Boumphrey, Sam Dooley, Shane McCarron, Sebastian Schnitzenbaumer & Ted Wugofski (2006). XHTML™ Modularization 1.1. W3C Working draft. <http://www.w3.org/TR/xhtml-modularization/> (May 2007)
- Anderson, John R. (1995). *Learning and Memory - An Integrated Approach*. New York: John Wiley.
- Andrews, Keith & Andreas Dieberger (1996). Reinventing the Wheels? Usability problems on the World-Wide Web. *Unpublished paper*. http://homepage.mac.com/juggle5-WORK/publications/Usability_and_Web.html (May 2007).
- Aula, Anne, Natalie Jhaveri & Mika Käki (2005). Information search and re-access strategies of experienced web users. In *Proceedings of WWW '05: 14th international conference on World Wide Web*: ACM Press, pages 583-592.
- Aula, Anne & Mika Käki (2003). Understanding Expert Search Strategies for Designing User-Friendly Search Interfaces. In *Proceedings of IADIS International Conference WWW/Internet 2003*, Vol. 2: IADIS Press, pages 759-762.
- Baeza-Yates, Ricardo A. & Berthier Ribeiro-Neto (1999). *Modern Information Retrieval*. Boston, MA: Addison-Wesley Longman.
- Baldi, Pierre, Paolo Frasconi & Padhraic Smyth (2003). *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*. New York: John Wiley and Sons.
- Barthes, Roland (1990). *S/Z*. Oxford: Blackwell. [Paris: Éditions du Seuil, 1970].
- Bederson, Benjamin B. (2002). A Structured 2D Graphics Framework (JAZZ Homepage). <http://www.cs.umd.edu/hcil/jazz/> (May 2007)
- Bederson, Benjamin B. (2004). Interfaces for Staying in the Flow - Computers should help us concentrate on our work, without concentrating on the computer. *Ubiquity* 5(27): 1.
- Bederson, Benjamin B. (2006). Piccolo in Comparison: Piccolo and Jazz (Website). <http://www.cs.umd.edu/hcil/piccolo/learn/comparison.shtml#jazz> (May 2007)
- Bederson, Benjamin B. (2007). FREE NoteLens 1.07 (for Windows). Windsor Interfaces, Inc. <http://www.windsorinterfaces.com/notelens.shtml> (May 2007)
- Bederson, Benjamin B. & Angela Boltman (1999). Does Animation Help Users Build Mental Maps of Spatial Information? In *Proceedings of Information Visualization Symposium (InfoVis 99)*, New York: IEEE Computer Society Press, pages 28-35.
- Bederson, Benjamin B., Aaron Clamage, Mary P. Czerwinski & George G. Robertson (2004). DateLens: A fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction (TOCHI)* 11(1): 90-119.
- Bederson, Benjamin B., James D. Hollan, Ken Perlin, Jonathan Meyer, David Bacon & George W. Furnas (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing* 7: 3-31.
- Bederson, Benjamin B. & Britt McAlister (1999). Jazz: An Extensible 2D+Zooming Graphics Toolkit in Java. Technical Report HCIL-99-07, CS-TR-4015, UMIACS-TR-99-24, Computer Science Department, University of Maryland, College Park, MD. <ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/99-07html/-99-07.html> (May 2007)
- Bederson, Benjamin B., Jon Meyer & Lance Good (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proceedings of User Interface and Software Technology (UIST)*: ACM Press, pages 171-180.

- Bederson, Benjamin B., Ben Shneiderman & Martin Wattenberg (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics* 21(4): 833 - 854. [HCIL-2001-18, CS-TR-4277, UMIACS-TR-2001-57, ISR-TR-2005-22].
- Bederson, Benjamin B., Larry Stead & James D. Hollan (1994). Pad++: Advances in Multiscale Interfaces. In *Proceedings of Human Factors in Computing Systems (CHI 94)*. ACM Press, pages 315-316.
- Belkin, Nicholas J. (1980). Anomalous state of knowledge as a basis for information retrieval. *Canadian Journal of Information Science* 5: 133-143.
- Belkin, Nicholas J., Pier Giorgio Marchetti & Colleen Cool (1993). BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval. *Information Processing & Management* 29(3): 325-344.
- Bellotti, Victoria & Yvonne Rogers (1997). From web press to web pressure: Multimedia representations and multimedia publishing. In *Proceedings of ACM SIGCHI'97 Conference on Human Factors in Computing Systems*, Atlanta, Georgia, pages 279-286.
- Berendt, Bettina, Bamshad Mobasher, Miki Nakagawa & Myra Spiliopoulou (2002). The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. In *Proceedings of ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2002) - 4th WebKDD 2002 Workshop*, Edmonton, Alberta, pages 159-179.
- Berglund, Anders (2006). Extensible Stylesheet Language (XSL) Version 1.1. <http://www.w3.org/TR/xsl> (May 2007)
- Bergman, Michael K. (2001). The deep web: Surfacing hidden value (White Paper). JEP - The Journal of Electronic Publishing, University of Michigan Press. <http://www.press.umich.edu/jep/07-01/bergman.html> (May 2007)
- Bernard, Michael L. & Barbara S. Chaparro (2000). Searching Within Websites: A Comparison of Three Types Of Sitemap Menu Structures. In *Proceedings of The Human Factors and Ergonomics Society*, San Diego, pages 441-444.
- Berners-Lee, Timothy J. (1989). Information Management: A Proposal. CERN (Conseil Européenne pour la Recherche Nucleaire). <http://www.w3.org/History/1989/proposal.html> (May 2007)
- Berners-Lee, Timothy J. (1996). The World Wide Web: Past, Present and Future. W3C. <http://www.w3.org/People/Berners-Lee/1996/ppf.html> (May 2007)
- Berners-Lee, Timothy J. (1997). Commentary on Web Architecture - Links and Law. W3C Design Issues. <http://www.w3.org/DesignIssues/LinkLaw.html> (Aug 2007)
- Berners-Lee, Timothy J. (2003). Press FAQ - What is the difference between the Net and the Web? <http://www.w3.org/People/Berners-Lee/FAQ.html#General1> (May 2007)
- Berners-Lee, Timothy J. (2006a). Das Netz der Zukunft wird unser Leben grundlegend verändern. *Interview by Thomas Kuhn*. Wirtschaftswoche (39) 2006. <http://www.wiwo.de/pswiwo/fn/ww2/sfn/buildww/id/2178/id/215308/fm/o/artpage/1/artprint/o/SH/3580025e35c22a0900c41cc4ab6468/depot/o/index.html> (May 2007)
- Berners-Lee, Timothy J. (2006b). Give yourself a URI. Timbl's Blog. <http://dig.csail.mit.edu/breadcrumbs/node/71> (May 2007)
- Berners-Lee, Timothy J. (2007). Testimony of Sir Timothy Berners-Lee (MIT) Before the United States House of Representatives: Hearing on the 'Digital Future of the United States: Part I - The Future of the World Wide Web'. <http://dig.csail.mit.edu/2007/03/01-ushouse-future-of-the-web.html> (May 2007)

- Berners-Lee, Timothy J. & Mark Fischetti (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco: Harper.
- Berners-Lee, Timothy J., James Hendler & Ora Lassila (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May 17, 2001. http://www.ryerson.ca/~dgrimsha/courses/cps720_02/resources/Scientific%20American%20The%20Semantic%20Web.htm (May 2007)
- Bernstein, Mark (1988). The bookmark and the compass: Orientation tools for hypertext users. *ACM SIGOIS Bulletin* 9(4): 34-35.
- Bertel, Sven (2001). Benutzerunterstützung im World Wide Web mit Hilfe räumlicher Konzepte. Diploma Thesis, Department for Informatics, Universität Hamburg, Hamburg. <http://www.cosy.informatik.uni-bremen.de/staff/bertel/publications/-diplomarbeit.pdf> (May 2007)
- Bieber, Michael, Fabio Vitali, Helen Ashman, Venkatraman Balasubramaniam & Harri Oinas-Kukkonen (1997). Fourth Generation Hypermedia: Some Missing Links for the World Wide Web. *International Journal of Human Computer Studies* 47(1): 31-65.
- Bieber, Michael & Jangling Wan (1994). Backtracking in a multiple window hypertext environment. In *Proceedings of Fifth ACM Conference on Hypermedia Technologies (ECHT'94)*, Edinburgh, pages 158-166.
- Bitner, James R. & Edward M. Reingold (1975). Backtrack programming techniques. *Communications of the ACM* 18(11): 651-656.
- Bolter, Jay David (1991a). Topographic Writing: Hypertext and the Electronic Writing Space. In *Hypermedia and Literary Studies*, edited by P. Delany & G. P. Landow. Cambridge: MIT Press, pages 105-118.
- Bolter, Jay David (1991b). *Writing Space: the Computer, Hypertext, and the History of Writing*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bos, Bert, Håkon Wium Lie, Chris Lilley & Ian Jacobs (1998). Cascading Style Sheets, Level 2 CSS2 Specification. W3C Recommendation. <http://www.w3.org/TR/REC-CSS2/> (May 2007)
- Broder, Andrei, Steve Glassman, Mark Manasse & Geoffrey Zweig (1997). Syntactic clustering of the Web. In *Proceedings of Sixth International World Wide Web Conference*, pages 391-404.
- Bruce, Harry, William Jones & Susan Dumais (2004). Keeping and Re-Finding Information on the Web: What Do People Do And What Do They Need? In *Proceedings of the 67th ASIST annual meeting (ASIST 2004)*, Chicago, IL, USA: Information Today, Inc.
- Bush, Vannevar (1945). As We May Think. *The Atlantic Monthly* 176(1): 101-108. [Reprinted in: *ACM Interactions*, Volume 3, Issue 2 (March 1996), pages 35-46; and in: *A History of Personal Workstations*. Edited by Adele Goldberg. New York: ACM Press (1988), pages 237-247; citations in this thesis refer to the page numbers of the ACM reprint].
- Byrne, Michael D., Bonnie E. John, Neil S. Wehrle & David C. Crow (1999). The tangled Web we wove: a taskonomy of WWW use. In *Proceedings of Conference on Human Factors in Computing Systems (CHI' 99)*, Pittsburgh, Pennsylvania: ACM Press, pages 544-551.
- Callahan, Jack, Don Hopkins, Mark Weiser & Ben Shneiderman (1988). An empirical comparison of pie vs. linear menus. In *Proceedings of ACM Conf. on Human Factors in Computing System (CHI 88)*, pages 95-100.
- Card, Stuart, K., Jock D. Mackinlay & Ben Shneiderman (1999). *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann.

- Card, Stuart K., George G. Robertson & Jock D. Mackinlay (1991). The Information Visualizer, an Information Workspace. In *Proceedings of Human Factors in Computing Systems (CHI 91)*: ACM Press, pages 181-188.
- Card, Stuart K., George G. Robertson & William York (1996). The Webbook and the Web Forager: An Information Workspace for the World Wide Web. In *Proceedings of Human Factors in Computing Systems (CHI 96)*: ACM Press, pages 111-117.
- Catledge, Lara D. & James E. Pitkow (1995). Characterizing Browsing Strategies in the World-Wide Web. In *Proceedings of Third International Conference on the World Wide Web*.
- Chi, Ed H. (2002). Improving Web Usability Through Visualization. *IEEE Internet Computing* 6(2): 64-71.
- Choo, Chun Wei, Brian Detlor & Don Turnbull (1998). A Behavioral Model of Information Seeking on the Web: Preliminary Results of a Study of How Managers and IT Specialists Use the Web. In *Proceedings of 61st ASIS Annual Meeting*, Vol. 35, Pittsburgh, PA: Information Today, pages 290-302.
- Choo, Chun Wei, Brian Detlor & Don Turnbull (2000a). Information Seeking on the Web: An Integrated Model of Browsing and Searching. *First Monday: Peer-Reviewed Journal on the Internet*: http://firstmonday.org/issues/issue5_2/choo (May 2007)5(2).
- Choo, Chun Wei, Brian Detlor & Don Turnbull (2000b). *Web Work: Information Seeking and Knowledge Work on the World Wide Web*. Dordrecht, Netherlands: Kluwer Academic Publishers.
- Cockburn, A. & S. Jones (1996). Which Way Now? Analysing and Easing Inadequacies in WWW Navigation. *International Journal of Human-Computer Studies* 45(1): 105-129.
- Cockburn, Andy, Saul Greenberg, Steve Jones, Bruce McKenzie & Michael Moyle (2003). Improving web page revisitation: Analysis, design and evaluation. *IT & Society* 1(3): 159-183.
- Cockburn, Andy & Bruce McKenzie (2001). What Do Web Users Do? An Empirical Analysis of Web Use. *International Journal of Human-Computer Studies* 54(6): 903-922.
- Combs, Tammara T. A. & Benjamin B. Bederson (1999). Does Zooming Improve Image Browsing? In *Proceedings of Digital Library (DL 99)*: ACM Press, pages 130-137.
- ComScore (2007). Worldwide Internet Audience has Grown 10 Percent in Last Year. <http://www.comscore.com/press/release.asp?press=1242> (May 2007)
- Conklin, Jeff (1987). Hypertext: a Survey and Introduction. *IEEE Computer* 20(9): 17-41.
- Connolly, Dan, Rohit Khare & Adam Rifkin (1997). The Evolution of Web Documents: The Ascent of XML. *World Wide Web Journal Special Issue on XML* 2(4): 119-128.
- Cooley, Robert, Bamshad Mobasher & Jaideep Srivastava (1999). Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems* 1(1): 5-32.
- Cooper, William S. (1971). A definition of relevance for information retrieval. *Information Storage and Retrieval* 7(1): 19-37.
- Counihan, Roger, Justin M. DeVoge & Ellen J. Bass (2005). A Web Architecture to support the tailoring of training content and learner assessment. In *Proceedings of IEEE Systems and Information Engineering Design Symposium*, Charlottesville, VA.
- Crow, Daniel & Barbara Smith (1992). DB_Habits: Comparing Minimal Knowledge and Knowledge-Based Approaches to Pattern Recognition in the Domain of User-Computer Interactions. In *Neural Networks and Pattern Recognition in Human-Computer Interaction*, edited by R. Beale & J. Finley. New York, NY: Ellis Horwood, pages 39-63.

- Csikszentmihalyi, Mihaly (1991). *Flow: The Psychology of Optimal Experience*. New York: Harper Perennial.
- Cugini, John & Jean Scholtz (1999). VISVIP: 3D Visualization of Paths Through Websites. In *Proceedings of International Workshop on Web-Based Information Visualization (WebVis 99)*, Florence, Italy, pages 259-263.
- Cutrell, Edward, Daniel C. Robbins, Susan Dumais & Raman Sarin (2006). Fast, flexible filtering with Phlat – Personal search and organization made easy. In *Proceedings of CHI 2006*, pages 261-270.
- Czerwinski, Mary, Eric Horvitz & Edward Cutrell (2001). Subjective Duration Assessment - An Implicit Probe for Software Usability. In *Proceedings of IHM-HCI 2001*, Lille, France, pages 167-170.
- Czerwinski, Mary P., Maarten van Dantzich, George G. Robertson & Hunter Hoffman (1999). The Contribution of Thumbnail Image, Mouse-over Text and Spatial Location Memory to Web Page Retrieval in 3D Viewing. In *Proceedings of IFIP Interact '99*, Edinburgh, Scotland: IOS Press, pages 163-170.
- Daft, Richard L. & Karl E. Weick (1984). Toward a Model of Organizations as Interpretation Systems. *Academy of Management Review* 9(2): 284-295.
- Daniel, Florian, Maristella Matera, Francesca Rizzo & ACM Press (2006). Extended memory (xMem) of web interactions. In *Proceedings of the 6th International Conference on Web Engineering*, Palo Alto, California, USA, pages 177-184.
- Darken, Rudolph Poul & John L. Sibert (1996a). Navigating large virtual spaces. *International Journal of Human-Computer Interaction* 8(1): 49-71.
- Darken, Rudolph Poul & John L. Sibert (1996b). Wayfinding strategies and behaviors in large virtual worlds. In *Proceedings of CHI'96*, New York: ACM Press, pages 142-149.
- Dave, Pratik, Unmil P. Karadkar, Richard Furuta, Luis Francisco-Revilla, Frank Shipman, Suwendu Dash & Zubin Dalal (2003). Browsing intricately interconnected paths. In *Proceedings of the fourteenth ACM Conference on Hypertext and Hypermedia*, Nottingham, UK, pages 95-103.
- De Lisa, Gene (1999a). How to drag and drop with Java 2, Part 1. JavaWorld online journal. <http://www.javaworld.com/javaworld/jw-03-1999/jw-03-dragndrop.html> (May 2007)
- De Lisa, Gene (1999b). Java Drag and Drop FAQ. Rockhopper Technologies, Inc. <http://www.rockhoppertech.com/> (May 2007)
- Dencker, Klaus Peter (1997). *INTERFACE 3: Labile Ordnungen. Netze denken - Kunst verkehren - Verbindlichkeiten*. Hamburg: Kulturbehörde Hamburg.
- Dervin, Brenda (1983). An overview of sense-making research: Concepts, methods and results to date. In *Proceedings of International Communications Association Annual Meeting*, Dallas, Texas, USA.
- Dhamija, Rachna & Adrian Perrig (2000). Déjà Vu: A user study using images for authentication. In *Proceedings of 9th USENIX Security Symposium*, Denver, Colorado, USA: USENIX Association.
- Dieberger, Andreas (1997). Supporting social navigation on the World Wide Web. *International Journal of Human-Computer Studies* 46: 805-825.
- Dodge, Martin & Rob Kitchin (2001). *Atlas of Cyberspace*. Amsterdam: Addison-Wesley Longman.
- Dömel, Peter (1994). WebMap - A Graphical Hypertext Navigation Tool. In *Proceedings of Second International WWW Conference*.
- Douglas, J. Yellowlees (1994). 'How Do I Stop This Thing?' Closure and Indeterminacy in Interactive Narratives. In *Hyper/Text/Theory*, edited by George P. Landow. Baltimore: Johns Hopkins UP, pages 159-188.

- Dumais, Susan (2007). Information Retrieval In Context. In *Proceedings of 12th International Conference on Intelligent User Interfaces*, Honolulu, Hawaii, USA, page 2.
- Dumais, Susan, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin & Daniel C. Robbins (2003). Stuff I've Seen - A System for Personal Information Retrieval and Re-Use. In *Proceedings of SIGIR 2003*, pages 72-78.
- Dunckel, Heiner, Walter Volpert, Martina Zölch, Ulla Kreutner, Cordula Pleiss & Karin Hennes (1993). *Kontrastive Aufgabenanalyse im Büro. Der KABA-Leitfaden - Grundlagen und Manual*. Stuttgart: Teubner.
- Eades, Peter (1984). A Heuristic for Graph Drawing. *Congressus Numerantium* 42: 149-160.
- Eckerson, Wayne W. (1995). Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications. *Open Information Systems* 10(1).
- Edwards, Deborah M. & Lynda Hardman (1989). Lost in hyperspace: cognitive mapping and navigation in a hypertext environment. In *Hypertext: Theory and Practice*, edited by Ray McAleese. Oxford, England: Intellect Books, pages 105-125.
- Eklund, John, James Sawers & Romain Zeiliger (1999). NESTOR Navigator: A Tool for the Collaborative Construction of Knowledge Through Constructive Navigation. In *Proceedings of Fifth Australian World Wide Web Conference (AusWeb 99)*, Lismore, pages 396-408.
- Ellis, David (1989). A Behavioural Model for Information Retrieval System Design. *Journal of Information Science* 15(4/5): 237-247.
- Ellis, David, Deborah Cox & Katherine Hall (1993). A Comparison of the Information Seeking Patterns of Researchers in the Physical and Social Sciences. *Journal of Documentation* 49(4): 356-369.
- Ellis, David & Merete Haugan (1997). Modelling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment. *Journal of Documentation* 53(4): 384-403.
- Engel, Frits L., J. J. Andriessen & H. J. R. Schmitz (1983). What, where and whence: Means for improving electronic data access. *International Journal of Man-Machine Studies* 18(2): 145-160.
- Engelbart, Douglas C. (1962). Augmenting Human Intellect: A Conceptual Framework. Summary Report AFOSR-3233 under Contract AF 49(638)-1024, SRI-Project 3578 for Air Force Office of Scientific Research., SRI (Stanford Research Institute), Menlo Park, CA. <http://sloan.stanford.edu/mousesite/EngelbartPapers/-Contents.html> (May 2007)
- Engelbart, Douglas C. (1963). A Conceptual Framework for the Augmentation of Man's Intellect. In *Vistas in Information Handling*, Vol. 1, edited by P. Howerton. Washington, DC: Spartan Books, pages 1-29. [Reprinted in Greif, Irene (ed.), 1988. *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo, California: Morgan Kaufmann Publishers, pages 35-65].
- Engelbart, Douglas C. & William K. English (1968). A Research Center for Augmenting Human Intellect. In *Proceedings of Fall Joint Computer Conference (AFIPS 1968)*, Vol. 33, San Francisco, CA, pages 395-410 (AUGMENT, 3954).
- Facca, Federico M. & Pier L. Lanzi (2005). Mining interesting knowledge from weblogs: a survey. *Data & Knowledge Engineering* 53(3): 225-241.
- Fitter, M. (1979). Towards more 'natural' interactive systems. *International Journal of Man-Machine Studies* 11(3): 339-350.
- Fitts, Paul M. & James R. Peterson (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47: 381-391.

- Fitts, Paul M. & James R. Peterson (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology* 67: 103-112.
- Floyd, Christiane (1984). A Systematic Look at Prototyping. In *Approaches to Prototyping*, edited by R. Budde. Berlin: Springer.
- Freksa, Christian & Christopher Habel (1990). Warum interessiert sich die Kognitionsforschung für die Darstellung räumlichen Wissens? In *Repräsentation und Verarbeitung räumlichen Wissens, Vol. 245 "Informatik Fachberichte"*, edited by C. Freksa & C. Habel: Springer, Berlin, pages 1-15.
- Fu, Xin, Tom Ciszek, Gary M. Marchionini & Paul Solomon (2005). Annotating the Web: An Exploratory Study of Web Users Needs for Personal Annotation Tools. In *Proceedings of the 68th Annual Meeting of the American Society for Information Science & Technology (ASIS&T)*, Vol. 42, Charlotte, NC, USA.
- Furnas, George W. (1986). Generalized Fisheye Views. In *Proceedings of CHI'86: Human Factors in Computing Systems*: ACM Press, pages 16-23.
- Furuta, Richard, Frank M. III Shipman, Catherine C. Marshall, Donald Brenner & Hao-wei Hsieh Hypertext Paths and the World-Wide Web: Experiences with Walden's Paths. In *Proceedings of Hypertext '97: the Eighth ACM Conference on Hypertext*, Southampton, U.K., pages 167-176.
- Gamma, Erich, Richard Helm, Ralph Johnson & John Vlissides (1996). *Entwurfsmuster (Design Patterns)*. Bonn: Addison-Wesley.
- Gandhi, Rajiv, Girish Kumar, Benjamin B. Bederson & Ben Shneiderman (2000). Domain Name Based Visualization of Web Histories in a Zoomable User Interface. In *Proceedings of 11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pages 591-598.
- Gilles, James & Robert Cailliau (2000). *How the Web was Born*. Oxford: Oxford University Press.
- Gillie, T. & D. Broadbent (1989). What makes interruptions disruptive? A study of length, similarity and complexity. *Psychological Research* 50: 243-250.
- Gomita (2007). Scrapbook - Firefox extension. <https://addons.mozilla.org/firefox/427/> (May 2007)
- González, Victor M. & Gloria Mark (2004). 'Constant, constant, multi-tasking craziness': managing multiple working spheres. In *Proceedings of SIGCHI conference on Human factors in computing systems*, Vienna, Austria, pages 113-120.
- Google (2006). Web History (former: Search History).
- Google (2007). Unternehmensprofil - Einige Zahlen. Google Inc. <http://www.google.de/intl/de/corporate/> (May 2007)
- Gordon-Murnane, Laura (2006). Social Bookmarking, Folksonomies, and Web 2.0 Tools. Red Orbit. http://www.redorbit.com/news/technology/541192/social_bookmarking_folksonomies_and_web_20_tools/index.html?source=r_technology (May 2007)
- Gourley, David & Brian Totty (2002). *HTTP: The Definitive Guide*. Cambridge, Tokyo: O'Reilly.
- Greenberg, Saul (1993). *The Computer User as Toolsmith - The Use, Reuse, and Organization of Computer-based Tools*: Cambridge University Press.
- Greenberg, Saul & Andy Cockburn (1999). Getting Back to Back: Alternate Behaviors for a Web Browser's Back Button. In *Proceedings of Fifth Conference on Human factors and the Web*.
- Greenberg, Saul, Geoffrey Ho & Shaun Kaasten (2000). Contrasting Stack-Based and Recency-Based Back Buttons on Web Browsers. Technical Report 2000-666-18, University of Calgary, Department of Computer Science, Calgary. http://pharos.cpsc.ucalgary.ca/Dienst/UI/2.0/Describe/ncstrl.ucalgary_cs/2000-666-18 (May 2007)

- Greenberg, Saul & Ian H. Witten (1988). How users repeat their actions on computers: Principles for design of history mechanisms. In *Proceedings of Human Factors in Computer Systems (CHI '88)*. Association for Computing Machinery, pages 171-178.
- Grøn­bæk, Kaj, Niels O. Bouvin & Lennert Sloth (1997). DHM/WWW- Integrating De­vise Hypermedia and the Web. Demonstration at Hypertext 97. In *Proceedings of The Eighth ACM International Conference on Hypertext.*, Southampton, UK.
- Grøn­bæk, Kaj & Randall H. Trigg (1999). *From Web to Workplace. Designing Open Hypermedia Systems*, Edited by Edward Barret. Cambridge, Mass. & London, Eng­land: MIT Press.
- Guernsey, Lisa (2004). Now Where Was I? New Ways to Revisit Web Sites. The New York Times. <http://www.nytimes.com/2004/01/22/technology/circuits/22next.html?ex=1390194000&en=fc394945f0f548c6&ei=5007&partner=USERLAND> (May 2007)
- GVU (1998a). Gvu's 10th WWW User Survey: Favorites/Bookmarks Usage Patterns. Gvu's WWW Surveying Team edition. Georgia Tech, Atlanta, GA, USA. http://www.gvu.gatech.edu/user__surveys/survey-1998-10/preview/use/q71.htm (May 2007)
- GVU (1998b). Gvu's 10th WWW User Survey: Home. Gvu's WWW Surveying Team edition. Georgia Tech, Atlanta, GA, USA. http://www.gvu.gatech.edu/user__surveys/survey-1998-10/ (May 2007)
- GVU (1998c). Gvu's 10th WWW User Survey: Problems Using the Web. Gvu's WWW Surveying Team edition. Georgia Tech, Atlanta, GA, USA. http://www.gvu.gatech.edu/user__surveys/survey-1998-10/graphs/use/q11.htm (May 2007)
- Hacker, Winfried (1986). *Arbeitspsychologie. Psychische Regulation von Arbeitstätigkeiten*. Berlin: VEB Deutscher Verlag der Wissenschaften.
- Hacker, Winfried & Peter Richter (1980). *Psychologische Bewertung von Arbeitsgestaltungsmaßnahmen - Ziele und Bewertungsmaßstäbe. Spezielle Arbeits- und Ingenieurpsychologie, Lehrtext 1*. Berlin: Deutscher Verlag der Wissenschaften.
- Halasz, Frank G. & Mayer Schwartz (1990). The Dexter Hypertext Reference Model. In *Proceedings of NIST Hypertext Standardization Workshop*, Gaithersburg, MD: National Institute of Standards, pages 95-133.
- Hammwöhner, Rainer (1997). *Offene Hypertextsysteme: das Konstanzer Hypertextsystem (KHS) im wissenschaftlichen Kontext*, Vol. 32. Konstanz: UVK, Universitäts-Verlag Konstanz.
- Heistermann, Horst (2007). WebWindow. [Since June 2001 replaced by the commercial successor product "WebWindow"]. <http://www.javio.com/> (May 2007)
- Herder, Eelco (2006). Forward, Back and Home Again - Analyzing User Behavior on the Web. Dissertation Thesis, University of Twente, Twente, The Netherlands.
- Herder, Eelco, Harald Weinreich, Hartmut Obendorf & Matthias Mayer (2006). Much to Know About History. In *Proceedings of Adaptive Hypermedia 2006, Lecture Notes in Computer Science*. Springer, pages 283-287.
- Hightower, Ron R., Laura T. Ring, Jonathan I. Helfman, Benjamin B. Bederson & James D. Hollan (1998a). Graphical Multiscale Web Histories: A Study of PadPrints. In *Proceedings of 9th ACM Conference on Hypertext and Hypermedia (Hypertext 98)*, Pittsburgh: ACM Press, pages 58-65.
- Hightower, Ron R., Laura T. Ring, Jonathan I. Helfman, Benjamin B. Bederson & James D. Hollan (1998b). PadPrints: Graphical Multiscale Web Histories. In *Proceedings of User Interface and Software Technology (UIST 98)*. ACM Press.
- Hill, William C. & James D. Hollan (1993). History-Enriched Digital Objects. In *Proceedings of Computers, Freedom and Privacy (CFP 93)*.

- Hill, William C., James D. Hollan, Dave Wroblewski & Tim McCandless (1992). Edit Wear and Read Wear. In *Proceedings of CHI'92*, Monterey, CA, USA: ACM Press, pages 3-9.
- Hirtle, Stephen C. & P. Bryan Heidorn (1993). The Structure of Cognitive Maps: Representation and Processes. In *Behaviour and Environment: Psychological and Geographical Approaches*, edited by T. Gärling & R.G. Golledge: Elsevier Science, Amsterdam, pages 170-192.
- Hong, Jason I. & James A. Landay (2001). WebQuilt: A Framework for Capturing and Visualizing the Web Experience. In *Proceedings of the 10th international conference on World Wide Web*, Hong Kong, Hong Kong, pages 717-724.
- Honkala, Mikko (2000). Using XML to Develop Applications for WAP and WWW Environments. M.Sc. Thesis, Department of Computer Science and Engineering, Helsinki University Of Technology, Helsinki.
- Höök, Kristina, Alan Munro & David Benyon (2003). *Designing Information Spaces: The Social Navigation Approach*. Springer.
- Hopkins, Don (1991). The design and implementation of pie menus. *Dr. Dobb's Journal* 16(12): 16-26.
- Horne, J. H. & Tom Lupton (1965). The work activities of 'middle' managers - An exploratory study. *The Journal of Management Studies* 2: 14-33.
- Hu, Yunhua, Hang Li, Yunbo Cao, Dmitriy Meyerzon, Li Teng & Qinghua Zheng (2006). Automatic Extraction of Titles from General Documents using Machine Learning. *Information Processing and Management* 42(5): 1276-1293.
- Hudson, James M., Jim Christensen, Wendy A. Kellogg & Thomas Erickson (2002). 'I'd be overwhelmed, but it's just one more thing to do:' Availability and interruption in research management. In *Proceedings of CHI 2002*: ACM Press, pages 97-104.
- Huebsch, Ryan, Joseph M. Hellerstein, Nick Lanham, Boon Thau Loo, Scott Shenker & Ion Stoica (2003). Querying the Internet with PIER. In *Proceedings of the 29th International Conference on Very Large Databases (VLDB)*, Berlin, Germany, pages 321-332.
- Huynh, David François (2004a). Mozilla + Firefox extension development: Calling Java code in custom JARs from Javascript. <http://mozilla-firefox-extension-dev.blogspot.com/2004/11/calling-java-code-in-custom-jars-from.html> (May 2007)
- Huynh, David François (2004b). SIMILE Project - Java Firefox Extension. <http://simile.mit.edu/java-firefox-extension/> (May 2007)
- IBM (2007). Web Intermediaries. <http://www.almaden.ibm.com/cs/wbi> (May 2007)
- Inforce_Bookmarks (2007). Inforce Bookmarks-Verwaltung. <http://www.inforce.de/Solutions/BookmarksMan.html> (May 2007)
- Jansen, Bernard J., Amanda Spink & Tefko Saracevic (2000). Real Life, Real Users and Real Needs: A Study and Analysis of User Queries on the Web. *Information Processing and Management* 36(2): 207-227.
- Johnson, Ralph (2004). Model-View-Controller as an Aggregate Design Pattern. <http://c2.com/cgi/wiki?ModelViewControllerAsAnAggregateDesignPattern> (May 2007)
- Jones, William, Harry Bruce & Susan Dumais (2001). Keeping found things found on the web. In *Proceedings of CIKM 2001*, pages 119-126.
- Jones, William, Harry Bruce & Susan Dumais (2003). How Do People Get Back to Information on the Web? How Can They Do It Better? In *Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)*, Zürich, Switzerland.
- Jones, William, Susan Dumais & Harry Bruce (2002). Once found, what then? A study of 'keeping' behaviors in personal use of Web information. In *Proceedings of the 65th ASIST Annual Meeting (ASIST 2002)*, Philadelphia, PA, USA, pages 391-402.

- Jones, William, Charles F. Munat, Harry Bruce & Austin Foxley (2005). The Universal Labeler: Plan the project and let your information follow. In *the American Society for Information Science and Technology (ASIST 2005)*, Vol. 42.
- Jones, William P. (2004). Finders, keepers? The present and future perfect in support of personal information management. First Monday. http://www.firstmonday.dk/issues/issue9_3/jones/index.html (May 2007)
- Jones, William P. & Susan T. Dumais (1986). The spatial metaphor for user interfaces: experimental tests of reference by location versus name. *ACM Transactions on Information Systems (TOIS)* 4(1): 42-63.
- Joyce, Michael (1991). Storyspace as a hypertext system for writers and readers of varying ability. In *Proceedings of Hypertext '91*, San Antonio, TX, pages 381-387.
- Joyce, Michael (1992). Afternoon, a story. 3rd edition. Eastgate Press, Cambridge, MA. <http://www.eastgate.com/catalog/Afternoon.html> (May 2007)
- Joyce, Michael (1995). *Of Two Minds: Hypertext Pedagogy and Poetics*.: University of Michigan Press.
- Jupiter_Research (2006). Web browsing beats page-turning. Jupiter Research as cited by BBC News. <http://news.bbc.co.uk/2/hi/business/6034433.stm> (May 2007)
- Kaasten, Shaun & Saul Greenberg (2001). Integrating Back, History and Bookmarks in Web Browsers. In *Proceedings of ACM Conference of Human Factors in Computing Systems (CHI'01 - Extended Abstracts)*, pages 379-380.
- Kabra, Govind, Chengkai Li & Kevin Chen-Chuan Chang (2005). Query Routing: Finding Ways in the Maze of the DeepWeb. In *Proceedings of the ICDE Workshop on Challenges in Web Information Retrieval and Integration (ICDE-WIRI 2005)*, Tokyo, Japan.
- Keep, Christopher, Tim McLaughlin & Robin Parmar (1993). The Electronic Labyrinth. <http://www.iath.virginia.edu/elab/hfloo37.html> (May 2007)
- Koffka, Kurt (1935). *Principles of Gestalt psychology*. New York: Harcourt, Brace, & World.
- Kosslyn, Stephen M., Thomas Ball & Brian J. Reiser (1978). Visual Images Preserve Metric Spatial Information: Evidence from Studies on Image Scanning. *Journal of Experimental Psychology: Human Perception and Performance* 4: 47-60.
- Krikelas, James (1983). Information seeking behavior: patterns and concepts. *Drexel Library Quarterly* 19: 5-20.
- Kurtenbach, Gordon & William Buxton (1993). The limits of expert performance using hierarchic marking menus. Proceedings of the SIGCHI conference on Human factors in computing systems. In *Proceedings of Conference on Human Factors in Computing Systems*, Amsterdam, The Netherlands, pages 482-487.
- Lai, Patrick & Udi Manber (1991). Flying Through Hypertext. In *Proceedings of Hypertext 1991*, San Antonio, Texas, USA: ACM Press, pages 123-132.
- Lakoff, G. & M. Johnson (1980). *Metaphors We Live By*. The University of Chicago Press.
- Lamping, John, Ramana Rao & Peter Pirolli (1995). A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of CHI 95 - Conference on Human Factors in Computing Systems*: ACM Press.
- Landow, George P. (1992). The Definition of Hypertext and Its History as a Concept. Chicago: John Hopkins University Press, pages 3-4.
- Landow, George P. (1994a). What's a Critic to Do? Critical Theory in the Age of Hypertext. In *Hyper/Text/Theory*, edited by George P. Landow. Baltimore, pages 1-48.
- Landow, George P. (1997). *Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology*. London, UK: The John Hopkins University Press.
- Landow, George W. (1994b). *Hyper/text/theory*. Baltimore: Johns Hopkins Univ. Press.
- Lansdale, Mark (1988). The psychology of personal information management. *Applied Ergonomics* 19(1): 55-66.

- Lawrence, Steve & C. Lee Giles (1999). Accessibility of Information on the Web. *Nature* 400: 107-109.
- Lee, Allison (1992). Investigations into History Tools for User Support. Ph. D. Thesis, Computer Systems Research Institute, University of Toronto, Toronto.
- Leontjew, A. N. (1974). The problem of activity in psychology. *Soviet Psychology* 13: 4-33.
- Leventhal, Michael, David Lewis & Matthew Fuchs (1998). *Designing XML Internet Applications*. Upper Saddle River, NJ: Prentice Hall.
- Lie, Håkon Wium & Janne Saarela (1999). Multipurpose Web Publishing Using HTML, XML and CSS. *Communications of the ACM* 42(10): 95-101.
- Liebscher, Peter & Gary M. Marchionini (1988). Browse and analytical search strategies in a full text CD-ROM encyclopedia. *School Library Media Quarterly* 7: 223-233.
- Linxi, Cai & A. Nico Habermann (1986). A history mechanism and undo/redo/reuse support in ALOE. Technical Research Report No. CMU-CS-86-148, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.
- Lynch, Kevin (1962). *The Image of the City*, Vol. 2nd. Cambridge: The M.I.T. Press & Harvard University Press.
- Lynch, Patrick J. & Sarah Horton (2002a). Online Style. In *Web Style Guide*. Yale Univ Press, pages 143-153.
- Lynch, Patrick J. & Sarah Horton (2002b). Site Design. In *Web Style Guide*. Yale Univ Press, pages 37-81.
- Macropool_Web-Recherche (2007). Web-Recherche-Netzwerkerweiterung. macropool. <http://www.macropool.de/de/produkte/web-recherche/erweiterungen/netzwerk.-html> (May 2007)
- MacWarriors, Special Interest Group of ACM - (2004). TrailBlazer. <http://www.acm.-uiuc.edu/macwarriors/projects/trailblazer/> (May 2007)
- Maglio, Paul P. & Teenie Matlock (1998). Constructing Social Spaces in Virtual Environments: Metaphors We Surf the Web By. In *Proceedings of Workshop on Personal and Social Navigation in Information Space*, Stockholm, Sweden, pages 138-149.
- Marchionini, Gary M. (1995). *Information Seeking in Electronic Environments*. Cambridge, England: Cambridge University Press.
- Marchionini, Gary M. & Ben Shneiderman (1988). Finding facts vs. browsing knowledge in hypertext systems. *IEEE Computer* 21(1): 70-80.
- Markwell, John & David W. Brooks (2003). 'Link Rot' limits the usefulness of Web-based educational materials in biochemistry and molecular biology. *Biochemistry and Molecular Biology Education* 31(1): 69-72.
- Marshall, Catherine C. (1998). Toward an ecology of hypertext annotation. In *Proceedings of the ninth ACM Conference on Hypertext and Hypermedia: links, objects, time and space - structure in hypermedia systems*, Pittsburgh, Pennsylvania, USA: ACM Press, pages 40-49.
- Marshall, Catherine C. & A. J. Bernheim Brush (2004). Exploring the relationship between personal and public annotations. In *Proceedings of the 4th ACM/IEEE-CS Joint International Conference on Digital Libraries*, Tuscon, AZ, USA: ACM Press, pages 349-357.
- Marshall, Catherine C., Frank G. Halasz, Russell A. Rogers & William C. Jr. Janssen (1991). Aquanet: a hypertext tool to hold your knowledge in place. In *Proceedings of Hypertext '91*: ACM Press.
- Marshall, Catherine C. & Frank M. Shipman (1993). Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext. In *Proceedings of Hypertext '93*, pages 217-230.
- Marshall, Catherine C., Frank M. Shipman & James H. Coombs (1994). VIKI: spatial hypertext supporting emergent structure. In *Proceedings of ACM European conference on Hypermedia technology September 1994*.

- Marston, Tony (2004a). Aren't the MVC and 3-Tier architectures the same thing? <http://www.tonymarston.net/php-mysql/infrastructure-faq.html> (May 2007)
- Marston, Tony (2004b). Implementing the 3-Tier architecture. <http://www.tonymarston.net/php-mysql/design-patterns.html#3tier.implementation> (May 2007)
- Mase, Motohiro & Seiji Yamada (2005). Extracting Topic Maps from Web browsing histories. In *Proceedings of The third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005)*, Singapore.
- Maurer, Hermann (1996). *Hyper-G, now Hyperwave: The Next Generation Web Solution*. United Kingdom: Longman Group.
- Mayer, Matthias (1997). n mal n minus eins halbe. Diploma Thesis, Hochschule für bildende Künste (HfbK), Hamburg, Germany. http://asi-www.informatik.uni-hamburg.de/personen/mayer/publications/m_mayer_diplom.pdf (May 2007)
- Mayer, Matthias (2000). Kontextvisualisierung: BrowsingIcons und BrowsingGraphs zur Verbesserung der Orientierung und Navigation im World Wide Web. In *Proceedings of 6. Tagung der Deutschen Sektion der Internationalen Gesellschaft für Wissensorganisation Hamburg, Germany*.
- Mayer, Matthias & Benjamin B. Bederson (2001). Browsing Icons: A Task-Based Approach for a Visual Web History. HCIL Technical Report: HCIL-200119, CS-TR-4308, UMIACS-TR-2001-85, HCI Lab, University of Maryland, MD, USA. <http://asi-www.informatik.uni-hamburg.de/personen/mayer/diss/papers/mayer-browsing-icons-uist2001.pdf> (May 2007)
- Mayo, Elton (1933). *The Human Problems of an Industrial Civilization*. Cambridge: Harvard University Press.
- McGrath, Sean (1998). *XML By Example*. Upper Saddle River, NJ: Prentice Hall.
- McKenzie, Bruce & Andy Cockburn (2001). An empirical analysis of web page revisitation. In *Proceedings of 34th Hawaiian International Conference on System Sciences, HICSS34*, Maui, Hawaii: IEEE Computer Society Press.
- Miller, Georg A., Eugene Galanter & Karl H. Pribram (1973). *Strategien des Handelns. Pläne und Strukturen des Verhaltens*. Stuttgart: Klett.
- Mintzberg, Henry (1970). Structured observation as a method to study managerial work. *The Journal of Management Studies* 7: 87-104.
- Mintzberg, Henry (1973). *The Nature of Managerial Work*. Englewood Cliffs: Prentice Hall.
- Misue, K., P. Eades, W. Lai & K. Sugiyama (1995). Layout Adjustment and the Mental Map. *Journal of Visual Languages and Computing* 6(2): 183-210.
- Miyata, Yoshiro & Donald A. Norman (1986). Psychological issues in support of multiple activities. In *User Centered System Design*, edited by D. A. Norman & S.W. Draper. Hillsdale, NJ: Lawrence Erlbaum, pages 265-284.
- Mobasher, Bamshad (2006). Data Mining for Personalization. In *The Adaptive Web: Methods and Strategies of Web Personalization*, Vol. 4321, *Lecture Notes in Computer Science*, edited by P. Brusilovsky, A. Kobsa & W. Nejdl. Berlin, Heidelberg: Springer-Verlag.
- Morkes, John & Jakob Nielsen (1997). Concise, SCANNABLE, and Objective: How to Write for the Web. Jakob Nielsen. <http://www.useit.com/papers/webwriting/writing.html> (May 2007)
- Moulthrop, Stuart (2006). A Subjective Chronology of Literary Hypertext. <http://iat-ubalt.edu/moulthrop/chrono.html> (May 2007)
- Mozilla__Project (2003). XML User Interface Language (XUL). <http://www.mozilla.org/projects/xul/> (May 2007)
- Mozilla__Project (2007). Firefox 2. www.mozilla.org/products/firefox (May 2007)

- MSDN (2007). MSDN: Internet Explorer Architecture. http://msdn.microsoft.com/workshop/browser/overview/ie_arch.asp (May 2007)
- Müller-Prove, Matthias (2002). Vision and Reality of Hypertext and Graphical User Interfaces. FBI-HH-B-237/02, Fachbereich Informatik (CS Department), Hamburg. <http://www.mprove.de/diplom> (May 2007)
- NCSA (1997). XMosaic's Hotlist. http://archive.ncsa.uiuc.edu/SDG/Software/XMosaic/UserGuide/XBook_31.html
- Nelson, Theodor Holm (1999). Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning and Deep Re-Use. *ACM Computing Surveys* 31(4es): Art. No. 37.
- Nelson, Theodor Holm (2003). I DON'T BUY IN. <http://ted.hyperland.com/buyin.txt> (May 2007)
- Nelson, Theodor Holm (2006). Project Xanadu. <http://www.xanadu.net/> (May 2007)
- Netcraft (2007). April 2007 Web Server Survey. Netcraft Ltd. http://news.netcraft.com/archives/web_server_survey.html (May 2007)
- Nielsen, Jakob (2005). Jakob Nielsen's Alertbox, January 3, 2005 - Reviving Advanced Hypertext. <http://www.useit.com/alertbox/20050103.html> (May 2007)
- Nielsen, Jakob (1990a). The Art of Navigating through Hypertext. *Communications of the ACM* 33(3): 296-310.
- Nielsen, Jakob (1990b). *Hypertext and Hypermedia*. Boston: Academic Press.
- Nielsen, Jakob (1995a). Jakob Nielsen's Alertbox for July 1995 - Features for the Next Generation of Web Browsers (Sidebar: Features Missing in Current Web Browsers). <http://www.useit.com/alertbox/newfeatures.html> (May 2007)
- Nielsen, Jakob (1995b). *Multimedia and Hypertext - The Internet and Beyond*. Boston: Academic Press.
- Nielsen, Jakob (1997a). Jakob Nielsen's Alertbox for July 15, 1997 - Search and You May Find. <http://www.useit.com/alertbox/9707b.html> (May 2007)
- Nielsen, Jakob (1997b). Jakob Nielsen's Alertbox for October 1, 1997 - How Users Read on the Web. Jakob Nielsen. <http://www.useit.com/alertbox/9710a.html> (May 2007)
- Nielsen, Jakob (2000). *Designing Web Usability: The Practice of Simplicity*, 2nd edition. Indianapolis, Indiana: New Riders Publishing.
- Nielsen, Jakob (2001). Jakob Nielsen's Alertbox, August 19, 2001 - Did Poor Usability Kill E-Commerce? <http://www.useit.com/alertbox/20010819.html> (May 2007)
- Nielsen, Jakob (2003). Jakob Nielsen's Alertbox, August 25, 2003 - Usability 101: Introduction to Usability. <http://www.useit.com/alertbox/20030825.html> (May 2007)
- Nielsen, Jakob (2004). Jakob Nielsen's Alertbox, May 3, 2004 - Change the Color of Visited Links. <http://www.useit.com/alertbox/20040503.html> (May 2007)
- Nievergelt, Jürg (1983). Die Gestaltung der Mensch-Maschine-Schnittstelle. In *GI 13. Jahrestagung*. Berlin: Kupka I., pages 41-50.
- Nievergelt, Jürg & Jean Weydert (1980). Sites, modes, and trails: Telling the user of an interactive system where he is, what he can do, and how to get places. In *Methodology of Interaction*, edited by R. A. Guedj, P. ten Hagen, F. R. Hopgood, et al. Amsterdam: North-Holland, pages 327-338.
- Norman, Donald A. (1993). *Things That Make Us Smart*. Reading, Mass.: Addison-Wesley.
- O'Reilly, Tim (2005). What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.
- O'Conaill, Brid & David Frohlich (1995). Timespace in the workplace: Dealing with interruptions. In *Proceedings of CHI 95*: ACM Press, pages 262-263.
- Obendorf, Hartmut, Harald Weinreich & Thorsten Haß (2004). Automatic Support for Web User Studies with SCONE and TEA. In *Proceedings of CHI '04: Conference*

- on *Human Factors in Computing Systems*: ACM Press Wien, Austria, pages 1135-1138.
- Obendorf, Hartmut, Harald Weinreich, Matthias Mayer & Eelco Herder (2007). Web Page Revisitation Revisited: Implications of a Long-term Click-stream Study of Browser Usage. In *Proceedings of CHI 2007*: ACM Press.
- Oberquelle, Horst (1994). Formen der Mensch-Computer-Interaktion. In *Einführung in die Software-Ergonomie. Gestaltung graphisch-interaktiver Systeme: Prinzipien, Werkzeuge, Lösungen*, edited by Eberleh, Edmund, Oberquelle, et al. Berlin, New York: Walter de Gruyter.
- Oesterreich, Rainer & Walter Volpert (1991). *VERA- Version 2, Handbuch und Manual*: TU Berlin.
- OneStat (2006). Global usage share Mozilla Firefox has increased according to OneStat.com. OneStat.com. http://www.onestat.com/html/aboutus_pressbox44-mozilla-firefox-has-slightly-increased.html (Oct 2006)
- Onestat (2007). Microsoft's Internet Explorer 7 Marketshare (Feb 2007). http://www.onestat.com/html/aboutus__pressbox50-microsoft-internet-explorer-7-usage.html (May 2007)
- Orlowski, Andrew (2002). Feel My Pain! – mapping the mind of Chairman Bill. The Register. http://www.theregister.co.uk/2002/04/23/feel_my_pain_mapping/ (May 2007)
- Palmer, S. E. & I. Rock (1994). Rethinking perceptual organization: The role of uniform connectedness. *Psychonomic Bulletin and Review*1(1): 29-55.
- Panko, Raymond R. (1992). Managerial communication patterns. *Journal of Organizational Computing* 2(1): 95-122.
- Parunak, H. Van Dyke (1991). Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning. In *Proceedings of Hypertext '91*, New York: ACM Press, pages 233-242.
- Peacock, Ian (1998). What is. a URI? <http://www.ariadne.ac.uk/issue18/what-is/> (May 2007)
- Pellegrini, Tassilo & Andreas Blumauer (2006). *Semantic Web. Wege zur vernetzten Wissensgesellschaft*. Berlin: Springer Verlag.
- Pemberton, Steve, et al. (2000). XHTML™ 1.0: The Extensible HyperText Markup Language, A Reformulation of HTML 4.0 in XML 1.0. W3C Recommendation, W3C. <http://www.w3.org/TR/xhtml1> (May 2007)
- Pierrakos, Dimitrios, Georgios Paliouras, Christos Papatheodorou & Constantine Spyropoulos (2003). Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction* 13(4): 311-372.
- Pirolli, Peter & Stuart K. Card (1995). Information foraging in information access environments. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pages 51-58.
- Pirolli, Peter & Stuart K. Card (1999). Information Foraging. *Psychological Review* 106(4): 643-675.
- Pitkow, James E. (1998). Summary of WWW Characterizations. *The Web Journal* 2: 3-13.
- Pitkow, James E. (2000). Summary of WWW Characterizations. In *Proceedings of the 7th International WWW Conference*, Brisbane, Australia.
- Polly, Jean Armour (1992). Surfing the Internet: An introduction. *Wilson Library Bulletin* 66(10): 38-42.
- Polly, Jean Armour (1994). Birth of a Metaphor - The Nascence of Surfing the Internet. <http://www.netmom.com/index.php?module=ContentExpress&func=print&ceid=3> (May 2007)

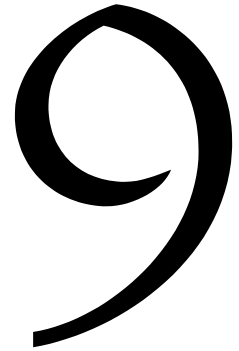
- Pomberger, Gustav & Günther Blaschek (1996). *Software Engineering. Prototyping und objektorientierte Software-Entwicklung*, 2nd edition. München: Carl Hanser Verlag.
- Porat, Marc U. (1977). *The Information Economy: Definition and Measurement*. US Department of Commerce, Office of Telecommunications, Washington DC.
- Pross, Harry (1972). *Medienforschung. Film, Funk, Presse, Fernsehen*. Darmstadt: Habel Verlag.
- Reenskaug, Trygve (2003). *The Model-View-Controller (MVC): Its Past and Present*. University of Oslo. http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC__-pattern.pdf (May 2007)
- Rizzo, Francesca, Florian Daniel, Maristella Matera, Sharon Albertario & Anna Nibioli (2006). Evaluating the semantic memory of web interactions in the xMem project. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, Venezia, Italy: ACM Press, pages 185-192.
- Robertson, George G., Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel & Maarten van Dantzich (1998). Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of User Interface and Software Technology (UIST 98)*. ACM Press, pages 153-162.
- Robertson, George G., Jock D. Mackinlay & Stuart K. Card (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of Human Factors in Computing Systems (CHI 91)*, New Orleans, Louisiana, USA: ACM Press, pages 189-194.
- Rosenberg, Jim (1993). *Intergrams*. Watertown MA: Eastgate Systems.
- Rosenberg, Jim (1996). The structure of hypertext activity. In *Proceedings of Hypertext'96*, pages 22-29.
- Russell, Daniel M., Mark J. Stefik, Peter Pirolli & Stuart K. Card (1993). The cost structure of sensemaking. In *Proceedings of Conference on Human Factors in Computing Systems ACM INTERCHI'93*: ACM Press, pages 269-276.
- Sandvad, Elmer S., Kaj Grønbaek, Lennert Sloth & Jørgen Lindskov Knudsen (2001). A metro map metaphor for guided tours on the Web: the Webwise guided tour system. In *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, Hong Kong, pages 326-333.
- Schefe, Peter (2001a). Hypertext I & II. In *"digital media"-lecture notes*. University of Hamburg, Hamburg, Germany.
- Schefe, Peter (2001b). Mediadesign - Introduction. In *"digital media"-lecture notes*. University of Hamburg, Hamburg, Germany.
- Schindler, Philipp (2006). Milliarden-Deal mit Youtube. Spiegel Online. <http://www.spiegel.de/netzwelt/netzkultur/0,1518,441876,00.html> (May 2007)
- Shen, Stewart N.T., Wing M. Kwok & Jacqueline Yang (1996). User-Centered, Usability-Based WWW Browser Facilities. Department of Computer Science, Old Dominion University. <http://www.cs.odu.edu/~shen/emowww5/Overview.html> (May 2007)
- Shepard, Robert N. & Jacqueline Metzler (1971). Mental Rotation of Three-Dimensional Objects. *Science* 171(3972): 701-703.
- Shipman, Frank M. III, Richard Furuta, Donald Brenner, Chung-Chi Chung & Hao-wei Hsieh (1998). Using paths in the classroom: experiences and adaptations. In *Proceedings of the ninth ACM Conference on Hypertext and hypermedia: links, objects, time and space-structure in hypermedia systems*, Pittsburgh, Pennsylvania, USA, pages 267-270.
- Shipman, Frank M. & Catherine C. Marshall (2000). Spatial Hypertext: An Alternative to Navigational and Semantic Links. *ACM Computing Surveys* 31(4).

- Shirai, Yoshinari, Yasuhiro Yamamoto & Kumiyo Nakakoji (2006). A history-centric approach for enhancing web browsing experiences. In *Proceedings of CHI '06 (extended abstracts) - Conference on Human Factors in Computing Systems*, Montréal, Québec, Canada: ACM Press, pages 1319-1324.
- Shneiderman, Ben (1992). Tree visualization with treemaps: a 2-d space-filling approach. *ACM Transactions on Graphics* 11(1): 92-99. [HCIL-91-03, CS-TR-2645, CAR-TR-548].
- Shneiderman, Ben (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of IEEE Visual Languages*, Los Alamitos: IEEE Computer Society Press, pages 336-343.
- Shneiderman, Ben (1998). *Designing the User Interface*, 3rd edition. Reading, MA: Addison-Wesley.
- Shneiderman, Ben & Greg Kearsley (1989). *Hypertext Hands-On! An Introduction to a new Way of Organizing and Accessing Information*. Reading, MA: Addison-Wesley.
- Siegel, A. & S. White (1975). The Development of Spatial Representation of Large-Scale Environments. In *Advances in Child Development and Behavior*, Vol. 10, edited by H.W. Reese. New York, NY: Academic Press, pages 9-75.
- Skopik, Amy & Carl Gutwin (2005). Improving revisitation in fisheye views with visit wear. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, Oregon, USA, pages 771-780.
- Slaughter, Laura A., Ben D. Harper & Kent L. Norman (1994). Assessing the Equivalence of Paper and On-Line Versions of the QUI5.5. In *Proceedings of 2nd Annual Mid-Atlantic Human Factors Conference*, pages 87-91.
- Spencer, Steven (1999). JavaWorld - Tip 66: Control browsers from your Java application. JavaWorld. <http://www.javaworld.com/javatips/jw-javatip66.html?011199txt> (May 2007)
- Spiliopoulou, Myra, Bamshad Mobasher, Bettina Berendt & Miki Nakagawa (2003). A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis. *INFORMSJ*. 15(2): 171-190.
- Spool, Jared & Erik Ojakaar (2001a). Are there users who always search? *User Interface Engineering Newsletter*.
- Spool, Jared & Erik Ojakaar (2001b). Users continue after category links. *User Interface Engineering Newsletter*.
- Sproull, Lee S. (1984). The nature of managerial attention. *Advances in Information Processing in Organizations* 1: 9-27.
- Stanton, Neville A. & Christopher Baber (1994). The Myth of Navigating in Hypertext: How a 'Bandwagon' Has Lost Its Course! *Educational Multimedia and Hypermedia* 3(3): 235-249.
- Stevens, Albert & Patty Coupe (1978). Distortions in Judged Spatial Relations. *Cognitive Psychology* 10: 422-437.
- Sun (2003). GraphLayout Examples. Sun Microsystems. <http://java.sun.com/products-/plugin/1.4.1/demos/applets.html> (Aug 2007)
- Tauscher, Linda (1996a). Evaluating History Mechanisms: An Empirical Study of Reuse Patterns in WWW Navigation. M.Sc. Thesis, Department of Computer Science, University of Calgary. <http://www.cpsc.ucalgary.ca/group/lab/papers/1996/96-Tauscher.Thesis/thesis.html> (Aug 2007)
- Tauscher, Linda (1996b). Supporting World Wide Web Navigation Through History Mechanisms. In *Proceedings of CHI 96 Workshop: HCI and the Web*, Vancouver, BC.

- Tauscher, Linda & Saul Greenberg (1996). Design Guidelines for Effective WWW History Mechanisms. In *Proceedings of Workshop on Designing for the Web: Empirical Studies Microsoft Corporation*.
- Tauscher, Linda & Saul Greenberg (1997a). How People Revisit Webpages: Empirical Findings and Implications for the Design of History Systems. *International Journal of Human-Computer Studies* 47(1): 97-137.
- Tauscher, Linda & Saul Greenberg (1997b). Revisitation patterns in World Wide Web navigation. In *Proceedings of Human factors in computing systems*, Atlanta, Georgia, USA, pages 399-406.
- TheCounter (2007). Browser Stats - April 2007. <http://www.thecounter.com/stats/2007-April/browser.php> (May 2007)
- Tiernan, Scott Lee, Shelly Farnham & Lili Cheng (2003). Two methods for auto-organizing personal web history. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '03)*, Ft. Lauderdale, Florida, USA: ACM Press, pages 814-815.
- Triesman, Anne & Stephen Gormican (1988). Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review* 95(1): 15-48.
- Trigg, Randall H. (1988). Guided tours and tabletops: tools for communicating in a hyper-text environment. *ACM Transactions on Office Information Systems* 6(4): 398-414.
- Trigg, Randall H. & M. Weiser (1986). TEXTNET: a network-based approach to text handling. *ACM Transactions on Office Information Systems* 4(1): 1-23.
- Tufte, Edward (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.
- Ulich, Eberhard (1983). Präventive Intervention im Betrieb: Vorgehensweisen zur Veränderung der Arbeitssituation. *Psychosozial* 6(20): 48-70.
- USC (2007). Center for the Digital Future: 2007 Digital Future Report. University of Southern California (USC) Annenberg School, Center for the Digital Future. http://www.digitalcenter.org/pages/current_report.asp?intGlobalId=19 (May 2007)
- USC, Center for the Digital Future (2004). The Digital Future Report: Surveying the Digital Future - Year Four - 2004. University of Southern California (USC) Annenberg School, Center for the Digital Future. <http://www.digitalcenter.org/downloads/DigitalFutureReport-Year4-2004.pdf> (May 2007)
- Validome (2007). Valide Webseiten nach W3C-Standard im deutschen Web. <http://www.-validome.org/lang/ge/html/valiwatch-web-2005-3> (May 2007)
- Volpert, Walter (1994). *Wider die Maschinenmodelle des Handelns. Aufsätze zur Handlungsregulationstheorie*. Lengerich, Berlin, Prag: Wolfgang Pabst Verlag.
- Volpert, Walter, Rainer Oesterreich, Silke Gablenz-Kolakovic, Tilmann Krogoll & Marianne E. Resch (1983). *Verfahren zur Ermittlung von Regulationserfordernissen in der Arbeitstätigkeit (VERA). Analyse von Planungs- und Denkprozessen in der industriellen Produktion*. Köln: TÜV Rheinland.
- W3C (1994). Uniform Resource Locators (URL). RFC 1738. Timothy J. Berners-Lee et al. <http://www.w3.org/Addressing/rfc1738.txt> (May 2007)
- W3C (1998). RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax. <http://www.ietf.org/rfc/rfc2396.txt> (May 2007)
- W3C (1999). Hypertext Transfer Protocol - HTTP/1.1. RFC 2616. <http://www.w3.org-Protocols/rfc2616/rfc2616.html> (May 2007)
- W3C (2000). A Little History of the World Wide Web. <http://www.w3.org/History.html> (May 2007)
- W3C (2001). XML Linking Language (XLink) Version 1.0. W3C Recommendation 27 June 2001. <http://www.w3.org/TR/xlink/> (May 2007)
- W3C (2002). Naming and Addressing: URIs, URLs. www.w3.org/Addressing (May 2007)

- W3C (2004). Architecture of the World Wide Web, Volume One - Identification. <http://www.w3.org/TR/webarch/#identification> (May 2007)
- W3C (2005). About the World Wide Web Consortium (W3C). <http://www.w3c.org/Consortium/> (May 2007)
- W3C (2006). The Ubiquitous Web Domain: Web access for anyone, anywhere, anytime, using any device. <http://www.w3.org/UbiWeb/> (May 2007)
- Ware, Colin (2000). *Information visualization: perception for design*. San Francisco, CA: Morgan Kaufmann Publishers Inc.
- WebBook_Video (1996). WebBook and Web Forager. The Open Video Project. <http://www.open-video.org/> (May 2007)
- Weick, Karl E. & Richard L. Daft (1983). The Effectiveness of Interpretation Systems. In *Organizational Effectiveness: A Comparison of Multiple Models*, edited by Kim S. Cameron & David A. Whetten. New York, NY: Academic Press, pages 71-93.
- Weinreich, Harald (2006). Welcome to Scone. <http://www.scone.de> (May 2007)
- Weinreich, Harald, Volkert Buchmann & Winfried Lamersdorf (2003). Scone: Ein Framework zur evaluativen Realisierung von Erweiterungen des Webs. In *Proceedings of KiVS 2003, Informatik Aktuell*, pages 31-42.
- Weinreich, Harald, Hartmut Obendorf & Eelco Herder (2006a). Data Cleaning Methods for Client and Proxy Logs. In *Proceedings of WWW 2006: Logging Traces of Web Activity - The Mechanics of Data Collection*, page 4.
- Weinreich, Harald, Hartmut Obendorf, Eelco Herder & Matthias Mayer (2006b). Off the Beaten Tracks: Exploring Three Aspects of Web Navigation. In *Proceedings of WWW Conference 2006*: ACM Press, pages 133-142.
- Weinreich, Harald, Hartmut Obendorf, Eelco Herder & Matthias Mayer (2007). Not Quite the Average: An Empirical Study of Web Use. *ACM Transactions on the Web (to appear)*.
- Weinreich, Harald, Hartmut Obendorf & Winfried Lamersdorf (2001). The Look of the Link Concepts for the User Interface of Extended Hyperlinks. In *Proceedings of Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext '01)*, Aarhus, Denmark: ACM Press.
- Weinreich, Harald, Hartmut Obendorf, Matthias Mayer & Eelco Herder (2006c). Der Wandel in der Benutzung des World Wide Webs. In *Proceedings of Mensch und Computer 2006*: Oldenbourg Wissenschaftsverlag, pages 155-164.
- Wen, James (2003). Post-valued recall web pages - User disorientation hits the big time. *IT & Society*1(2): 184-194.
- Wexelblat, Alan & Pattie Maes (1999). Footprints: History-Rich Tools for Information Foraging. In *Proceedings of Computer Human Interaction (CHI)*, pages 270-277.
- Wikimedia (2007). Image: Layout engine usage share.svg. Wikimedia Commons. en.wikipedia.org/wiki/Image:Layout_engine_usage_share.svg (04 2007)
- Wilson, P. (1973). Situational Relevance. *Information Storage and Retrieval* 9: 457-471.
- Woodruff, Allison, Andrew Faulring, Ruth Rosenholtz, Julie Morrision & Peter Pirolli (2001). Using thumbnails to search the Web. In *Proceedings of*, Seattle, Washington, USA: ACM Press, pages 198-205.
- Xu, Geoffrey, Andy Cockburn & Bruce McKenzie (2001). Lost on the Web: An Introduction to Web Navigation Research. In *Proceedings of the 4th New Zealand Computer Science Research Students Conference, 2001*.
- Xue, Yewei, Yunhua Hu, Guomao Xin, Ruihua Song, Shuming Shi, Yunbo Cao, Chin-Yew Lin & Hang Li (2007). Web Page Title Extraction and Its Application. *Information Processing and Management* (to appear).
- Yates, Francis A. (2001). *The Art of Memory*. Chicago: University Of Chicago Press.

- Youn, Hee Yong, Minkoo Kim & Hiroyuki Morikawa (2006). Ubiquitous Computing Systems: Third International Symposium, Ucs 2006 Seoul, Korea, October 11-13, 2006. Springer.
- Zakon, Robert Hobbes (2006). Hobbes' Internet Timeline v8.2. <http://www.zakon.org/~robert/internet/timeline/> (May 2007)
- Zellweger, Polle T. (1989). Scripted Documents: A Hypermedia Path Mechanism. In *Proceedings of Hypertext 89*, Baltimore: ACM Press, pages 1-14.
- Zheng, Qixing, Kellogg Booth & Joanna McGrenere (2006). Co-authoring with structured annotations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montréal, Québec, Canada: ACM Press, pages 131-140.



Appendices

And just as the Japanese amuse themselves by filling a porcelain bowl with water and steeping in it little crumbs of paper which until then are without character or form, but, the moment they become wet, stretch themselves and bend, take on colour and distinctive shape, become flowers or houses or people, permanent and recognisable, so in that moment all the flowers in our garden and in M. Swann's park, and the water-lilies on the Vivonne and the good folk of the village and their little dwellings and the parish church and the whole of Combray and of its surroundings, taking their proper shapes and growing solid, sprang into being, town and gardens alike, from my cup of tea.

(Marcel Proust, Swann's Way)

The first appendix, 9.1, provides a subjective statement that concludes this thesis from a personal perspective. All subsequent appendices directly refer to a certain chapter where they were referenced as sources for further detail. They do not follow a special order.

9.1 Personal Epilogue

The presented work can be seen as one chapter of an enduring personal fascination and engagement with human beings, their individual lives, and the exploration of



Figure 9-1: GPS visualization from the PHYSIS art project - two weeks' path.

This figure spans a timeframe of approximately two weeks in 1996, driving each day to the university (upper part), and once to my sister about an hour away (lower part), using two different highways. Discontinuities in the paths are caused by bad signal reception or the Elb-tunnel near Hamburg.

these lives' external traces. Paths on the web are just one example of such external traces.

For three years in the early 1990s, while studying at the school of fine arts in Hamburg, I carried a small “Garmin” GPS receiver with me day after day in order to trace all the paths that I took in the physical world – as part of an art project entitled PHYSIS. The paths included those created by cycling to the university in the morning, visiting a friend in the evening, or going on a trip over a weekend – Garmin was always with me. I wrote a small piece of software that rendered the raw data both into 2D and 3D space. These artificial visualizations had no applicatory purpose; instead, the idea was to use my body as a means for creating aesthetical pieces of imagery. The resulting images, many of them caused by the most trivial day-to-day activities, were aesthetical in themselves. I felt no need to move in salient patterns as do participants of posterior GPS drawing projects.¹ Figure 9-1 and Figure 9-2 show exemplary path visualizations created during the PHYSIS project.

Another example for the visualization of external traces is depicted on page vii showing a conversation via web pages (similar to a wiki web) of about twenty conference participants who discussed the topic of the emerging web in 1994.

¹ Such as found at <http://www.gpsdrawing.com/gallery.htm>

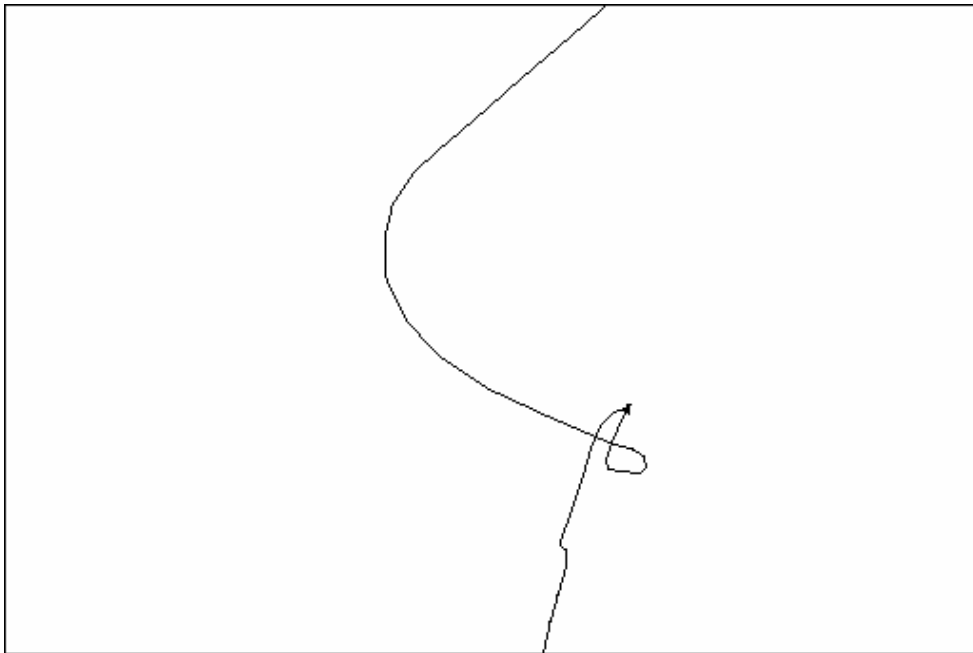


Figure 9-2: GPS visualization from *PHYSIS* art project - one afternoon. On the way to my sister, taking a highway exit. Zoomed into a detail in the figure above.

Such external traces of human beings may be interpreted metaphorically as externalizations of internal processes. They remind us about who we are and what our lives are about. They can be read as material complements to our lives' thread. According to Greek mythology, the three Moirae, personifications of destiny, are responsible to spin, measure, and cut this "thread of life". Figure 9-3 shows a piece I produced in 1990, entitled "Bungee / Thread of life", presenting a man, whose life is hanging by a thread. It reveals both visually and metaphorically some equivalence to the Moirae and their work, as well as to Ariadne in Theseus' Maze, relying on her potentially life-saving thread, to the surfer on the web, who is dependent on working history tools that enable him to refer to his past, and to the information worker in general, who is existentially dependent on technological aids. Our technological "extensions" become something like a connecting umbilical cord, that unites us with the world around, and that we use and need to understand and to adapt to our environment. A playful exploration of these thoughts can be found along the three theoretical positions represented by John R. Searle, Marshall McLuhan, and Oswald Wiener in my diploma thesis (1997).



Figure 9-3: "Bungee / Thread of life" (1990)
One of six images of an early piece by the author (photo emulsion, glass).

9.2 Technological Background

Several technical characteristics of the web are relevant in the context of revisitation support. Those that were of importance during this project will briefly be outlined in the following.

Due to its unexpected fast and uncontrolled growth, influenced by a heterogeneous environment of scientists, technologists, browser manufacturers, and the W3C (World Wide Web Consortium)¹ as standardization committee, the web does not provide a uniform technological basis, which provided several stumbling blocks during development. Many different solutions coexist, recommendations of the W3C are only partly implemented, and a broad variety of technical de-facto standards were introduced by the diverse browser brands and versions. The technical problems faced during the design and implementation of the prototype, such as the detection of frame pages and the control of multiple windows, were based on missing APIs and standards.

Figure 9-4 shows the five areas of possible standardization in the realm of (web) documents, as presented by Scheffe (2001a: chapter 1.6). The web offers helpful standards in some areas, but lacks them in others.

Content models that determine the inner structure of documents exist in a wide variety, provided by open or proprietary standards such as HTML, XML, XHTML, MP3, AAC, Vorbis, and PDF. As in several other cases, however, hardly any document found on the web is fully compliant to the related standard. According to (Validome 2007) only 3.9% of all German web pages conform to the W3C standards. The SessionGraphs prototype only parsed HTML documents, but could be easily extended to other document types. It would be a valuable next step to extend the history prototype to be able to reference, for example, also special positions within a PDF document or an audio or video file that could receive their own characteristic representations in a session visualization with increased functionality such as play, pause, forward, etc. A more monolithic approach, as for example postulated by the Dexter hypertext reference model (Halasz & Schwartz 1990)², has the advantage of specifying content models only at one central place, in this case in the within-component layer. On the other hand, monolithic approaches are much less scalable.

¹ International consortium of member organizations, full-time staff, and the public, with the mission to lead the web to its full potential by developing protocols and guidelines that ensure long-term growth for the web (W3C 2005).

² The Dexter Hypertext Reference Model is the outcome of two small workshops in 1988 that were attended by leading hypertext researchers. They were held in the Dexter Inn in New Hampshire hence the name for the model.

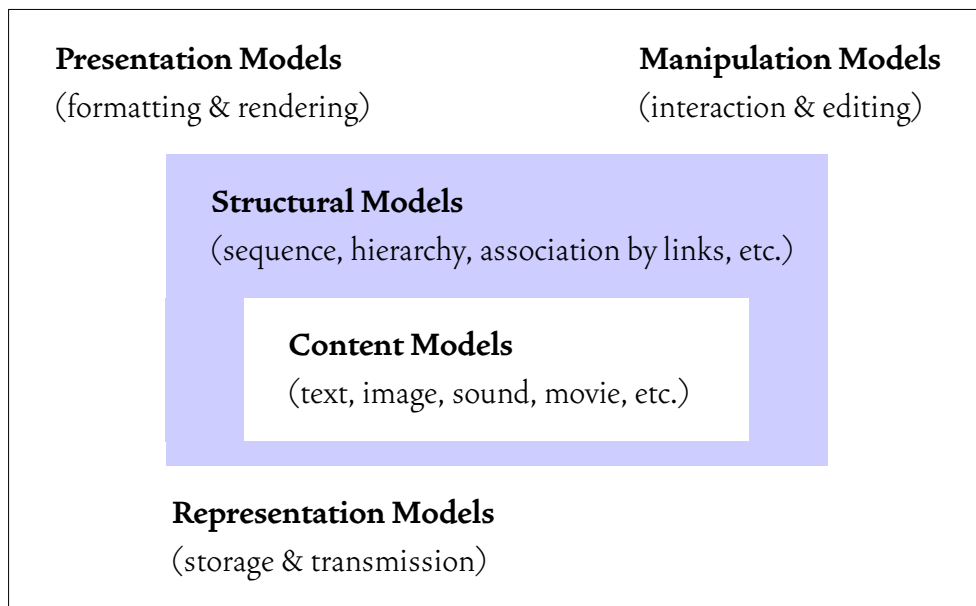


Figure 9-4: (Web) document models (Scheffe 2001a: chapter 1.6).

Structural models that establish the relationships *between* different (components of) documents such as associations, sequences, or hierarchies are still under-represented on the web – while the Dexter model, for example, provides the storage layer and anchoring interface to specify structures such as bidirectional links and links of arbitrary arity that may have multiple start and end points (Halasz & Schwartz 1990: 34). The XML Linking Language (XLink, version 1.0, a W3C recommendation W3C 2001) would at least extend the web by bidirectional links, which then also influenced revisitation behavior since links could be used instead of the back button. A history tool uses the possibilities provided by a structural model to define the actual references to documents.

Representation models are responsible for the storage and transmission of documents, managed in the Dexter model by the storage layer. The Web, however, has no representation model. The transport of resources is done transparently via the stateless Hypertext Transfer Protocol HTTP. It can be used for arbitrary content. Yet, it does not provide any means for versioning. Already in the late nineteen-sixties, Ted Nelson addressed in his Xanadu project (Nelson 2006, 1999) the necessity of versioning mechanisms in global hypertext systems as an essential concept to archive documents, avoid dangling links, and trace the original origin of information.

Intercepting the transfer of objects on the Web is easily possible by using intermediaries. They can be used to filter and monitor all transferred content. The Scone framework - used for the SGA prototype - comprises a programmable intermediary and other components to analyze and modify transferred web documents and to trace the user actions.

Presentation models that determine how content is rendered (Dexter: presentation specifications layer) and *manipulation models* that determine how the user can interact with it (Dexter: run time layer) are one of the weakest points of the web, since they are largely missing. Developments such as CSS (Cascading Style Sheets) and XSL (the eXtensible Style Language) define some aspects of a presentation model. XSL includes the transformation language XSLT and the formatting language XSL FO (XSL Formatting Objects) (Berglund 2006), both XML applications. XSLT allows to define rules to transform an XML document into another document type, for example an HTML document, while XSL FO allow to define the rendering of XML data on screen or paper, but also as sound. These techniques are still rarely used and undersupported by current browsers. Many issues are controlled by the browser software individually, just as the manufacturers believe they are beneficial. Thus, these techniques were not applied in the existing SessionGraphs prototype, but should become interesting for upcoming implementations.

Future web history approaches would certainly profit from an easy, standardized way to access internal browser representations, as well as from a commonly accepted definition for what exactly is understood by a history. This thesis tries to contribute to this goal by introducing a well-differentiating nomenclature and a first practical definition.

Encoding Web Pages – HTML and XML

The encoding of web pages was important in this project since certain elements, such as page title or number of outgoing links, had to be extracted from the documents and other elements, such as the Scone applet, had to be integrated into them. The most common language for web page encoding is *HTML 4.01*, an SGML (Standard Generalized Markup Language) application according to ISO 8879. As markup language, it allows to add both structural and semantic information explicitly to a document, thereby augmenting automatic processing abilities. Basic building blocks of a markup language are *elements* that can include the content and other elements, identified with start- and end-tags (McGrath 1998). An example (document with title-tag) is shown in Figure 9-5. One advantage is the distinction between form and content¹ – they do not control the final rendering of a document, thereby increasing device independence, searchability, and reusability (Honkala 2000: 4).

The aspired independence of structure and content is massively weakened by a shortcoming of HTML that provides no standard for defining the page layout. In a “browser war” between Netscape Navigator and Internet Explorer during HTML versions 2.0 to 3.2 – many elements were differently supported and interpreted. Elements such as tables were and still are misused for layout purposes; font attributes were directly stored within the documents, which is against the original goal of

¹ A clear distinction cannot always be made (for a discussion see Connolly, Khare & Rifkin 1997).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Hera's home page</TITLE>
  </HEAD>
  <BODY>
    <H1>Hera's infoviz home page</H1>
    <P>The next information visualization course starts on Mar 11.</P>
  </BODY>
</HTML>
```

Figure 9-5: A simple HTML document.

device independence. *Cascading Style Sheets* (CSS – since *HTML 4.0*) improved this situation notably. CSS2, a style sheet language recommended by the W3C, contains more than 100 powerful styling properties (Bos, Lie, Lilley & Jacobs 1998) such as colors, margins, and font attributes, but is not yet sufficiently supported. Changing a single style sheet, authors can change the appearance of their whole site. The development of a history tool has to face the current inhomogeneous situation, which is quite laborious and bears many possibilities of unwanted behavior that can only be abolished in a limited number of cases.

The *XML* (eXtensible Markup Language), a subset of SGML, is a meta-language for defining application-specific, tag-based languages to encode structured documents (Leventhal, Lewis & Fuchs 1998). XML preserves most of the power and richness of SGML, but reduces the overwhelming complexity drastically (Pemberton 2000). According to (Honkala 2000) it was also “designed to put an end once and for all to the tag-soup wars propagated by Microsoft and Netscape”.¹ XML and its related standards provide a much more flexible and standardized way to structure, interchange, and store documents than HTML. XML documents may refer to a DTD (Document Type Definition) that can specify further user- and domain-specific elements (Lie & Saarela 1999).

Currently, XHTML (the eXtensible HyperText Markup Language), a W3C-motivated reformulation of the common HTML 4.01 standard as XML 1.0 application (see Altheim, Boumphrey, Dooley, McCarron, Schnitzenbaumer & Wugofski 2006), is slowly replacing HTML. Providing a suite of expandable XML tag-sets, it allows to simply adding new elements to an existing DTD, thereby improving portability since alternative platforms, like mobile devices, often lack the computing power to accommodate ill-formed HTML as current web browsers do. While XHTML 1.0 was designed to be backwards compatible to most browsers, XHTML 1.1 is no longer.

¹ Jim Cape, in a post to comp.infosystems.www.authoring.html on June 3, 1997.

The outlined development shows that the web is on its path towards a much more prudent environment compared to the still existing solutions that include numerous flaws. Future history projects will be able to use these technologies that were not mature and widespread enough during the presented work.

Transferring Web Pages – HTTP

Objects on the web are traditionally transferred using the Hypertext Transfer Protocol (HTTP)¹, a stateless request/response protocol on the application layer for distributed, collaborative, hypermedia information systems (W3C 1999). It is based on a client-server architecture, in which the client sends the request to the server (including request method, URI, protocol version) causing the server to respond with a status line (protocol version, success or error code, MIME-like message that contains server information and possible entity-body content) (W3C 1999).

HTTP 1.1 defines, amongst others, the request methods POST and GET. GET allows retrieving whatever information is identified by a URI. The client (web browser) transmits the URI and receives the addressed object in response. POST allows sending additional data to the server that may be used for example to post a message on a bulletin board. Both methods are frequently used as a workaround to implement session tracking on the server side, since the stateless protocol does not support session tracking itself. Undoing multi-step transactions, which would be a valuable history functionality, requires additional server-based solutions. Since these solutions are left to the site providers, a multitude of different behaviors can be found so that the user cannot know what a certain action will effect – a major shortcoming of current web history. Since GET parameters are part of the URI, measurements of recurrence rates must exactly define how they deal with these parameters (see discussion on page 176).

Referencing Resources on the Web – URLs, URIs, URNs

Resources on the web are identified using URIs (Universal Resource Identifiers W3C 2004), which comprise both Uniform Resource Locators (URLs) and Uniform Resource Names (URNs). URLs are associated with popular URI schemes such as http, mailto, and ftp. URNs refer to URIs with an institutional commitment to persistence and availability. They are intended to serve as persistent, location-independent, resource identifiers (see also: W3C 1998, 2002, 1994) and Ian Peacock (1998).

In combination with history tools, URIs are essential for addressing the target represented in a history. Furthermore, an interesting question to be addressed in

¹ The term *web* as it is used in this thesis also includes object that are transferred with alternative protocols, such as ftp (File Transfer Protocol), mailto (electronic mail services using SMTP, POP, and IMAP protocols), news (USENET news), nntp (USENET news using NNTP access), irc (Internet Relay Chat), telnet (Reference to interactive sessions), wais (Wide Area Information Servers), file (Host-specific file names) and others.

the semantic web context would be whether a person's individual history should also become a first class object, i.e. given its own URI, thereby becoming referenceable, searchable, and processable. According to Tim Berners-Lee (2006b), "everything of importance deserves a URI", and so he suggests everybody to give oneself a URI, which is already possible and according to the FOAF (Friend-of-A-Friend) project also highly recommendable. Based on an appropriate ontology, an individual history could be defined as a part of a person and be represented in a machine-readable manner, referenced by its individual URI. This would open up completely new forms for uses of the web; users could formulate queries like "Tell me who I could have met half a year ago when I was in Venice for a week", or "Who also likes this book", or "Who has to write a term paper on this book?", or "Where do people like Marc prefer to go in London?" etc.

HTTP-URLs that had to be processed in this project have the form `http://<host>:<port>/<path>?<search-part>` (W3C 1994).¹ They identify web-based resources by naming the scheme ("http"), a machine name or address, and a path interpreted according to the scheme. The *host-part* of URLs determines the server that hosts the resource. It can either be a machine name that will be resolved to an IP address by a domain name server (DNS) or an IP address directly (a 4 byte value). Optionally, a port number can be specified. A valid name is, for example, "www.informatik.uni-hamburg.de:8088". In this case, the top level domain is "de"; "uni-hamburg" and "informatik" are two nested sub-domains, and "www" is the individual machine name. 8088 is the port number. A DNS resolves this string to the IP address 134.100.8.250.

The *path-part* is hierarchically structured, reflecting the structure of the web server directories where the resource is located. It may be followed by a filename and a fragment identifier, indicated by a leading hash sign ('#') that references a subregion within the referenced HTML object.

One URL specifies exactly one location on the web. A resource at a specific location may be modified or removed, potentially resulting in broken links. No mechanism guarantees the consistence between resources, links, and URLs. URLs are transient identifiers, which is a shortcoming of current web technology. Thus, many attempts to revisit pages in the long term are likely to result in error messages. The consistent availability of URNs would make long-term revisits much easier.

Linking and Embedding Web Pages

Hyperlinks that interconnect different HTML pages are created using the anchor element and are part of the referencing HTML document's source. The anchor element's HREF (hypertext reference) attribute allows to specify the target using a URL. In the target, the anchor's NAME attribute allows to specify a named destination. A link may look like ``

¹ In this document, the term URL always refers to HTTP-URLs.

ABC Group. Similarly, other objects such as images are embedded into HTML pages, e.g. like this: ``. In the presented prototype and in the implemented studies, the page source had to be parsed and the contained links were processed appropriately.

Nielsen (2005) points out, that the current web still lacks several important navigational features, such as “fat” (multi-tailed) and typed links. The XLink recommendation (W3C 2001) introduces more expressive power than HTML links, such as multiple link destinations, external linkbases and thus also overlapping links, typed links, etc. Current web browsers, however, still hesitate to support XLink. If implemented in browsers, they would make the synchronous visitation and revisitation of entire groups of related pages easier (see Nielsen 1990b: 109).

Revisitation and Technological Shortcomings

The described technological environment bears several shortcomings in relation to revisitations. First, there is no common agreement about what a history exactly is, how revisitations should be supported, and by which means. These decisions are left to the browser manufacturers, and are based on their estimations and knowledge. This also means that each browser shows different history tools, even if the main ones are implemented similarly. Currently, personal histories are hardly transferable to other browsers, to other computers, and hardly exchangeable with other people. They are still far from being machine-readable. Personal annotations, an important ingredient to a matured history environment, are also not commonly available.

Nielsen (2005) who also criticizes this situation is optimistic: “The last ten years were a black hole: much attention was focused on doomed attempts at making the Web more like television. Hopefully, the next decade will focus instead on empowering users and giving us the features we need to master a worldwide information space”.

9.3 Data Related to Prototypical Implementation

This appendix refers to the SessionGraphs prototype A described in chapter 3.

9.3.1 Classes Overview and Source Lines of Code (SLOC)

This appendix gives an overview over all 147 implemented classes and packages of the SessionGraphs “prototype A” in its final version. Source lines of code (SLOC) have been measured with the JSLOC tool developed by Brien Alkire (www.alkires.com/software/JSLOC.htm). In total, the code comprises 16,203 source lines of code.

Package	SLOC	source file	SLOC	Package	SLOC	source file	SLOC				
sga.config	2128	Config.java	334	sga.sessions.sessiongraphs (cont'd)		VisNodeContentSGCam.java	70				
		ConfigMain.java	207			LookAheadGraph.java	66				
		ConfigFileReader.java	167			VisNodeContentMiniatureMapCam.java	60				
		ConfigContentNodesTemplate.java	133			SGSelectionMoveHandler.java	44				
		ConfigContentRadioBtnsTemplate.java	118			MiniatureMouseAdapter.java	43				
		NodeSizeVerticalSliderTemplate.java	103			LookAheadMouseAdapter.java	35				
		ConfigContentMixedTemplate.java	90			ThumbGeneratorThread.java	33				
		ConfigFileWriter.java	83			SGLabelContentMiniatureMapCam.java	32				
		NodeSizeSliderTemplate.java	81			SGLabelContentSGCam.java	32				
		ColorTemplate.java	78			ThumbMouseMotionAdapter.java	31				
		NotYetAppliedDialog.java	54			MiniatureMapAnimator.java	25				
		ConfigContentLinksTemplate.java	50			SGAnimator.java	25				
		ConfigContentPositionerTemplate.java	50			SGPanEventHandler.java	17				
		ConfigContentRadioBtnsSimpleTemplate.java	50			MiniatureMouseMotionAdapter.java	15				
		InfoDialog.java	49			LookAheadMouseMotionAdapter.java	14				
		BooleanTemplate.java	46			MiniatureMapMouseAdapter.java	13				
		WholeNumberField.java	43			MySelectableVisualLeaf.java	12				
		DoubleTemplate.java	40			VisNodeContentIF.java	12				
		IntTemplate.java	40			MyDeepArrayList.java	11				
		StringTemplate.java	39			VisEdgeContentIF.java	11				
		ConfigContentBackgroundTemplate.java	36			MiniatureMapCamera.java	9				
		ConfigContentPanel.java	36			SGCamera.java	9				
		ConfigContentColorTemplate.java	28			MiniatureMapPositionerIF.java	7				
		ConfigContentSliderTemplate.java	27			PositionerIF.java	7				
		ConfigFileDeleter.java	26			IconListenerIF.java	4				
		BooleanObject.java	17			SGLabelContentIF.java	4				
		ConfigContentButtonTemplate.java	17								
		ColorObject.java	16			sga.tasks	203	TaskRepresentation.java	110		
		DoubleObject.java	16					SessionNode.java	30		
		IntObject.java	16					TaskTreeNode.java	25		
		StringObject.java	14					TaskNode.java	20		
		ConfigAttributeIF.java	9					TaskTreeModel.java	18		
		ConfigWindow.java	7								
		ConfigContentEmptyTemplate.java	5					sga.ui	2155	SplittedPaneFrame.java	469
		ConfigListenerIF.java	3							TaskChooser.java	435
										MyCellEditor.java	286
										MyCellRenderer.java	187
						MyStringTransferable.java	160				
						MyDnDTextTree.java	122				
						SessionView.java	103				
						MyDnDTree.java	87				
						TaskTreeMouseAdapter.java	79				
						TaskView.java	73				
						MyTreeTransferable.java	67				
		TChAndLookAheadPanel.java	54								
		TaskViewWindowMonitor.java	13								
		SplittedPaneMouseAdapter.java	12								
		SplittedPaneMouseMotionAdapter.java	8								
		sga.utils	1230	PrintUtils.java	287						
				DateUtils.java	123						
				DataDisplay.java	115						
				BrowserControl.java	72						
				NewTitleDialog.java	63						
				SessionGraphsEvaluator.java	63						
				UseTimeMeasureTool.java	61						
				CommentDialog.java	60						
				FATest.java	55						
				InfoDialog.java	55						
		SplashPopup.java	49								
		ThumbnailGenerator.java	38								
		SGnodePrintMA.java	30								
		FrameTimer.java	26								
		SGnodePrintMMA.java	19								
		MiniaturePrintMA.java	17								
		SavingWindowMonitor.java	15								
		SGPrintMA.java	15								
		MiniatureMapPrintMA.java	14								
		TaskTreePrintMA.java	14								
		BasicWindowMonitor.java	9								
		MiniaturePrintMMA.java	9								
		TaskTreePrintMMA.java	8								
		MiniatureMapPrintMMA.java	7								
		AbstractMeasurableSessionController.java	6								
		Total SLOC		16203							
sga.io	1799	SGAFileReaderMultiFiles.java	737								
		SGAFileReader.java	458								
		HistoryPersistenceManager.java	429								
		SGAFileWriterMultiFiles.java	155								
		ReaderIF.java	10								
		WriterIF.java	10								
		sga.log	1245	AutoLogger.java	1197						
				DayIntPair.java	24						
				DayLongPair.java	24						
		sga.model	2506	SessionController.java	1139						
				SGASconePlugin.java	410						
				CmdTable.java	355						
				WebObject.java	313						
				WebObjectVisit.java	221						
				Visit.java	40						
				FrameCapsule.java	28						
		sga.sessions	1023	DemoSessionGenerator.java	559						
				SessionRepresentation.java	290						
				DefaultSessionHeuristics.java	103						
				SessionHeuristicsIF.java	63						
DemoSurfer.java	8										
sga.sessions.sessiongraphs	3914	VisNode.java	490								
		SessionGraph.java	471								
		StoppingNodePositioner.java	221								
		SingleLookAheadGraph.java	195								
		VisNodeHubContentMiniatureMapCam.java	164								
		VisNodeHubContentSGCam.java	164								
		VisEdgeContentSGCam.java	151								
		VisEdgeContentMiniatureMapCam.java	143								
		SGSelectionModifyHandler.java	135								
		MiniatureMap.java	126								
		DefaultPlusNodePositioner.java	123								
		MiniatureGraph.java	120								
		VisNodeMouseAdapter.java	116								
		SGCompositeSelectionHandler.java	111								
		VisEdge.java	108								
		StoppingGlueNodePositioner.java	94								
		SGLabel.java	93								
		SmoothMiniatureMapPositioner.java	91								
		LookAheadGraphsPanel.java	81								
		VisNodeCopySelection.java	76								

Table 9-1: SessionGraphs code - all classes, source lines of code (SLOC)

9.4 Data Related to Two Controlled Experiments

This appendix refers to the experiments described in chapter 4.2.

9.4.1 Consent Form

Each participant signed the following consent form prior to participation:

1. I have volunteered to participate in this experiment.
2. I have been informed in advance as to what my task(s) would be and what procedures would be followed.
3. I have been given the ability to ask questions before the experiment, and have had my questions answered to my satisfaction.
4. I am aware that I have the right to withdraw consent and discontinue participation at any time, without prejudice.
5. My signature below may be taken as affirmation of all the above, prior to participation.

9.4.2 Experimental Tasks and Questionnaire

Figure 9-6 presents an exemplary computer screen, generated using a small Java application, that was used to present the questions to the study participants and to collect their answers. Also the measuring timer was driven by this interface. The entire text presented to the subjects, including tasks, comments, and questions, is reprinted thereafter. In the study it was all laid out and presented similarly to Figure 9-6. This reprint equals the questions as presented to one of the four study groups. For the other three groups the sequential order changed, while the content of the questions remained the same.

Screens from experiment I

[Initial slide:]

Thank you very much for helping us!

During the next hour we will do the following: First we will collect some data from you. Then you will have to solve certain tasks by surfing the web. For one set of tasks you will use Netscape in combination with the history tool we developed. The other set of tasks you will do without the tool. We will measure the time

it takes to complete each task. After the tasks we would like you to answer some questions regarding the usability of the two different systems.

Because we study a history tool it is essential to measure also how people revisit pages after longer periods of time. Therefore it is necessary that you come in next week again for some very short tasks (10 minutes). We will not measure you, but the system! You just can help us to find the pros and cons. Please do not tell other

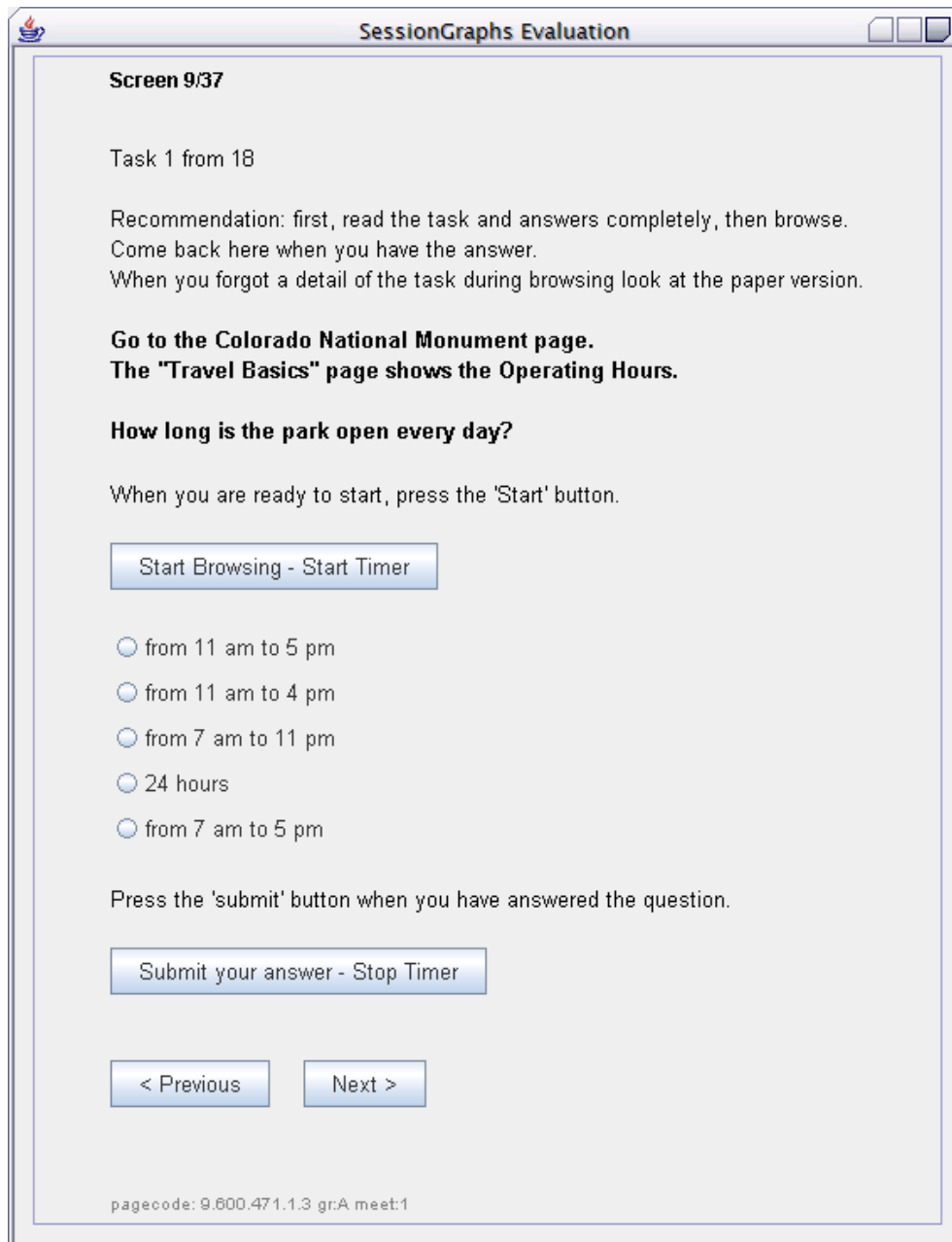


Figure 9-6: Exemplary study screen.

students about the specific tasks. We will exactly tell you when we start and stop to measure the time in each case.

Please answer the following questions so that we understand your background. All data will be anonymized before we publish it.

- Email address:
- Sex: Male, Female

- Age: < 20, 20-29, 30-39, 40-49, >= 50
- Student: (Y/N)
- Programmer: (Y/N)
- Major or Field:
- Computer use per week (hours): 0-2, 3-5, 6-9, 10-19, >= 20
- Browsing the web per week (hours): 0-2, 3-5, 6-9, 10-19, >= 20
- The web browser you use:
- Have you used Netscape before?

- Do you often use Favorites or Bookmarks?
- If so, do you organize them hierarchically?
- Do you often use the history list?
- When will you come next week for ten minutes?

You will now learn how to use SessionGraphs with four demo tasks.

Go to the Fort Washington Park page. Go to "Activities". On this page you find a picture. What does it show?

- A fort
- A cannon
- A forest
- A mountain
- visitors

Please go to the SessionGraphs and make a new SessionRepresentation now.

Go to the Catoctin Mountain Park page. How many camping grounds do they list?

- 1
- 2
- 3
- 4
- 5

Go back to the Fort Washington Park "Activities" page. What phone number is listed at the bottom?

- (301)763-4600
- (301)125-0880
- (301)156-3221
- (301)642-123
- (301)343-3423

Go back to the Catoctin Mountain Park Camping page. When is the Camp Misty Mount open?

- From 04/05/2000 To 10/28/2001
- From 04/10/2000 To 10/28/2001
- From 04/15/2000 To 10/28/2001
- From 04/25/2000 To 10/28/2001
- From 04/30/2000 To 10/28/2001

Now we begin with the real tasks.

Task 1 from 18

Recommendation: first, read the task and answers completely, then browse. Come back here when you have the answer. When you forgot a detail of the task during browsing look at the paper version.

Go to the Colorado National Monument page. The "Travel Basics" page shows the Operating Hours.

How long is the park open every day?

When you are ready to start, press the 'Start' button.

- from 11 am to 5 pm
- from 11 am to 4 pm
- from 7 am to 11 pm
- 24 hours
- from 7 am to 5 pm

Task 2 from 18

Go to the Sequoia & Kings Canyon National Parks page. The facilities page announces something new to "COMING SOON!".

What is it?

- GIANT TREE MUSEM
- GIANT FOREST MUSEM
- SPECIAL TOURS
- GENERAL GRANT TOUR
- FREE ENTRANCE

Please go to the SessionGraphs and make a new SessionRepresentation now.

Task 3 from 18

Visit the page of the Chesapeake & Ohio Canal National Historical Park.

Look at the fees. How much is a "Commercial Vehicle (1 to 6 Passengers)" for 3 days?

- \$25
- \$30
- \$35
- \$40
- \$45

Task 4 from 18

Visit the page of the Grand Canyon Park. Look at the fees. How much is a the "Entrance Fee" for a Vehicle for 7 days?

- \$5
- \$12

- \$15
- \$20
- \$25

Task 5 from 18

Now we begin to revisit some pages you already have visited.

Go back to the "Travel Basics" at Colorado National Monument. The "Weather and Climate" section says how warm the summers are. How warm are they?

- 60 degrees to 110 degrees
- 75 degrees to 100 degrees
- 85 degrees to 100 degrees
- 90 degrees to 110 degrees
- 80 degrees to 100 degrees

Task 6 from 18

Go back to the Facilities page of Sequoia & Kings Canyon National Parks.

Look at the section Foothills Visitor Center (in Sequoia).

What does the next line say?

- Open In Summer 8am-4:30pm
- Open All Year 8am-4:30pm
- Open In Summer 9am-5:30pm
- Open All Year 9am-5:30pm
- Open All Year 9am-4:00pm

Task 7 from 18

Regarding the Chesapeake & Ohio Canal National Historical Park: Where is the entrance fee charged for a "Commercial Vehicle (1 to 6 Passengers)"?

- Great Falls Area
- Little River Area
- Bay View Area
- Canal View Area
- Chesapeake River Area

Task 8 from 18

Regarding the Grand Canyon Park Fees: How much is the "Backcountry Impact Fee" per day?

- \$2
- \$3
- \$5
- \$6
- \$10

Task 9 from 18

Now compare the home pages of the following three parks. Each of the home

pages shows one photo of the park in the upper right.

- Grand Canyon Park
- Sequoia & Kings Canyon Nat. Parks
- Colorado National Monument

How many of the three pictures are colored and not black & white?

- 1
- 2
- 3

Now we change the tools. You can take a short break.

Task 10 from 18

Recommendation: first, read the task and answers completely, then browse.

Come back here when you have the answer. When you forgot a detail of the task during browsing look at the paper version.

Go to the Big Bend National Park.

The "Travel Basics" page shows the Operating Hours. How long is the park open every day?

When you are ready to start, press the 'Start' button.

- 24 hours daily, from April to October
- 24 hours daily, all year
- from 7 am to 11 pm, all year
- 24 hours daily, March until September
- from 7 am to 5 pm, all year

Task 11 from 18

Go to the Death Valley National Park page. Look at the "Lodging" information. What's the name of the first Inn listed?

- Gabriel's River Inn
- Sunshine Inn
- Furnace Creek Inn
- Death River Inn
- Sunny Inn

Task 12 from 18

Visit the page of the Yosemite National Park. Follow the link "Links" on the right. What's the phone number on the right side?

- 237-555-1233
- 375-782-3256
- 847-665-8976
- 456-656-8766
- 760-647-3044

Task 13 from 18

Visit the page of the Hot Springs National Park. Look at the Facilities.

When is the HOT SPRINGS NATIONAL PARK VISITOR CENTER open?

- Open All Year 8:00 a.m.-5:30 p.m.
- Open All Year 9:30 a.m.-4:30 p.m.
- Open All Year 9:00 a.m.-5:30 p.m.
- Open All Year 9:00 a.m.-5:00 p.m.
- Open All Year 10:00 a.m.-4:30 p.m.

Task 14 from 18

Now we begin to revisit some pages you already have visited.

Go back to the Big Bend National Park "Travel Basics" page. If you want to go there by car: How many "paved roads" lead there?

- 4
- 2
- 6
- 3
- none

Task 15 from 18

Go back to the Death Valley National Park "Lodging" page. What's the phone number of the "Panamint Springs Resort"?

- (775) 482-7680
- (771) 235-2330
- (711) 546-3566
- (771) 322-5444
- (711) 355-6556

Task 16 from 18

Revisit the "Links" page of the Yosemite National Park page. How many "NEARBY ATTRACTIONS" are listed?

- 1
- 2
- 3
- 4
- 5

Task 17 from 18

Look again at the Facilities page of the Hot Springs National Park. What's the phone number of the visitor center?

- 554-123-321
- 421-345-4684
- 823-543-9743
- 527-658-3257
- 501-624-3383

Task 18 from 18

Now compare the home pages of the following three parks. Each of the home pages shows one photo of the park in the upper right.

Big Bend National Park

Death Valley National Park

Yosemite National Park

How many of the three pictures are colored and not black & white?

- 1
- 2
- 3

Congratulations! You are done with all the tasks.

Next we would like to get some subjective impressions how you estimate both systems. After this we are done for today.

Question 1 from 6: Overall reactions.

Please try to find for each system (Netscape with Browsing Icons and Netscape alone) the most appropriate description.

Overall reactions: Netscape WITH SessionGraphs

[The following QUIS scales were offered:]

- | | |
|--------------------|------------------|
| terrible | - wonderful |
| frustrating | - satisfying |
| useless | - useful |
| difficult to use | - easy to use |
| difficult to learn | - easy to learn |
| inadequate power | - adequate power |

Overall reactions: Netscape alone - WITHOUT SessionGraphs

- | | |
|--------------------|------------------|
| terrible | - wonderful |
| frustrating | - satisfying |
| useless | - useful |
| difficult to use | - easy to use |
| difficult to learn | - easy to learn |
| inadequate power | - adequate power |

Room for additional comments regarding your overall impression:

Question 2 from 6

For answering these questions please consider just the revisiting support of each system. Again, please try to find for each system the most appropriate description.

Revisiting Pages: Netscape WITH SessionGraphs

- difficult - easy
- slow - fast
- with obstacles - direct
- confusing - clear
- frustrating - satisfying

Revisiting Pages: Netscape alone - WITHOUT SessionGraphs

- difficult - easy
- slow - fast
- with obstacles - direct
- confusing - clear
- frustrating - satisfying

Room for additional comments regarding the revisitation of pages:

Question 3 from 6: The visual design of history elements. Again, please try to find for each system the most appropriate description.

Visual Design: SessionGraphs (the large and small graphs, the tree)

- not useful - useful
- bad - good
- visually unclear - visually clear
- not pleasurable - pleasurable

Visual Design: Netscape (Backbutton, Bookmarks, History List)

- not useful - useful
- bad - good
- visually unclear - visually clear
- not pleasurable - pleasurable

Room for additional comments regarding the visual design:

Question 4 from 6

These questions just regard the SessionGraphs. Please give short catchwords or phrases.

- What is your general impression of the SessionGraphs tool?
- What are the most positive things about the tool?
- What are the most negative things about the tool?
- What did you miss?
- What would you suggest to improve?

Question 5 from 6

These questions just regard the SessionGraphs. How would you describe each aspect?

The shapes of the graphs

[QUIS scale:]

not useful for navigation ... useful for navigation

The animation (movement) of nodes.

bad ... good

The use of thumbnails.

not useful for navigation ... useful for navigation

The interaction with single nodes.

difficult ... easy

The search function.

not useful ... useful

The larger graphs with titles at each page (second part of window).

not useful ... useful

The smaller graphs without titles at each page (third part of window).

not useful ... useful

The task and session concept.

bad ... good

Room for additional comments regarding any of these aspects:

Question 6 from 6

This is the last questions page. After this you are done. **Do you have any additional comments?** Please think about it for a minute.

Thank you very much for your time!

You are now finished. We see us next week. Press the 'Finish' button to end this session.

Screens from experiment II

Thank you very much for coming again!

We will have again two sets of tasks, one with and one without the tool.

You will revisit some pages you already have visited in the first meeting.

After this we have a few more questions.

It should take about 10 minutes today.

Task 1 from 8

Go back to the Big Bend National Park "Travel Basics" page.

If you want to go there by car: How is the first "paved road" called which leads there?

- State Route 321 from Marathon
- U.S. 385 from Marathon
- State Route 421
- Ranch Road 125 from Ginger Bay
- U.S. 412

Task 2 from 8

Go back to the Death Valley National Park "Lodging" page. What's the URL of the "Furnace Creek Inn"?

- www.furnacecreekinn.com
- www.fcinn.com
- www.furnaceinn.com
- www.furnacecreekresort.com
- www.furnacecreek.com

Task 3 from 8

Revisit the "Links" page of the Yosemite National Park page. How many "LOCAL PARTNERS & INFORMATION RESOURCES" are listed?

- 1
- 2
- 3
- 4
- 5

Task 4 from 8

Now compare the home pages of the following three parks. Each of the home pages shows one photo of the park in the upper right.

Big Bend National Park

Death Valley National Park

Yosemite National Park

How many of the three pictures show persons on them?

- 0
- 1
- 2

Now we change the tools. You can take a short break.

Task 5 from 8

Go back to the "Travel Basics" at Colorado National Monument. What is the

name of the airport by which you can get there?

- Colorado Airport
- Colorado Junction Airport
- Junction Valley Airport
- Grand Junction Airport
- Grand Valley Airport

Task 6 from 8

Go back to the Facilities page of Sequoia & Kings Canyon National Parks.

Look at the section GRANT GROVE VISITOR CENTER.

What does the next line say?

- Open All Year
- Open All Year .Winter: 9 am - 4:30 pm
- Open All Year. Winter: 8 am - 5 pm
- Open All Year. Winter: 8 am - 3:30 pm
- Open All Year 8am-4:00pm

Task 7 from 8

Regarding the Chesapeake & Ohio Canal National Historical Park: How much is a "Cyclist/Walker" for 3 days?

- Free
- \$1
- \$2
- \$3
- \$5

Task 8 from 8

Now compare the home pages of the following three parks. Sequoia & Kings Canyon National Parks

Grand Canyon Park Colorado National Monument.

How many of them offer a link to "Learn More about the History of the Park"

- 1
- 2
- 3

Thank you!

You are done with all the tasks today. We have a few more questions. After that you are done.

Question 1 from 4

Which system did you like more to revisit pages after some days?. Please tell us why.

Question 2 from 4

How would you describe the experience of revisiting pages after some days using the two different tools?

Please try to find for each system the most appropriate description.

Revisiting Pages: Netscape WITH SessionGraphs

terrible	- wonderful
difficult	- easy
slow	- fast
not useful	- useful
with obstacles	- direct
confusing	- clear
frustrating	- satisfying

Revisiting Pages: Netscape alone - WITHOUT SessionGraphs

terrible	- wonderful
difficult	- easy
slow	- fast
not useful	- useful
with obstacles	- direct
confusing	- clear
frustrating	- satisfying

Room for additional comments regarding the revisitation of pages:

Question 3 from 4

Please give short explanations.

Compared to the first day: was it harder, easier or equally difficult to revisit pages? Why?

SessionGraphs:

Netscape alone:

For revisiting pages after some days: What was good about each tool? what was bad?

SessionGraphs:

Netscape alone:

Only for SessionGraphs:

Did you recognize the shape of your graphs again? What could we do to make them more recognizable?

Question 4 from 4

This is the last questions page. After this you are done. **Do you have any additional comments?** Please think about it for a minute.

Thank you very much for your time!

You are now finished. Press the 'Finish' button to end this session.

9.4.3 Participants' Comments

In the following, all written comments collected during the two experiments are presented. They were entered by the participants using the designated text fields (see previous chapter). No corrections are added, misspellings are sustained. Since the tool then still was named BrowsingIcons (also abbreviated as “BI” or “bi”), these terms will appear instead of SessionGraphs. Since some answers topically overlapped with several questions, they were added to the most appropriate question here. For questions that received many answers, the answers were grouped after the study according to common subtopics. The subtopics are emphasized using square brackets: “[...]”.

9.4.3.1 First Meeting – Short Term Revisits

The following text presents the phrases *exactly* as the participants typed them into the screen interface, including misspellings.

Comments on the general impression of the SessionGraphs tool

[*useful*]

- Pretty useful. I like the idea of visualizing using graphs.
- I think it is a great tool. It is much more intuitive than netscape's history mechanism.
- browsing icons was useful to navigate websites. Especially netscape cannot show history as trees so sometimes I cannot keep all histories of mine. But bi can do. That's great! Also it shows small thumbnails for pages, which is also helpful.
- It is great for providing direct access to previously visited pages. I did not get to use the overview/detail function of the graphs. It may be good for reviewing previous work in longer time periods. (as opposed to just navigation and revisitation)
- Browsing icons were well helpful to organize the previous visiting pages.
- [BI] was much faster and easier to access previous pages since they were all presented on the screen where I was browsing without having to open a separate drop-down list or window. Because of this I used this tool much more than the history tool fo Netscape. It also seemed faster to load pages.

[*nice tool with potential of improvement*]

- The graph is visually clear, but the labels are sometimes confusing
- netscape without the new tool -- satisfying? No it just is -- today's reality, or what we are stuck with -- the alternative was much more interesting to use -- I liked the concept of thumbnails for returning to the correct page, but in actual fact didn't find myself using them. If I know that I will revisit the home page, I will make use of the existing tools in Netscape effectively. When a session gets bigger, the graph can become confusing.
- It will become even better with some changes. I enjoyed using it.

[*fun*]

- Fun
- It was definitely fun -- and the immediate

feedback was appreciated.

- cool.
- fun
- Nice visualization of web history

[*areas of improvement*]

- I have to understand the graph in the BI. The graph is not that intuitive for the beginner.
- The BI tool was harder to learn since this was the first time I've seen it and had three or four connected windows.

[*additional comments*]

- Netscape only tool: Since the tasks took me back to second or third level pages, it was easier and faster to go back to the home page and redo the browsing than open any of the history tools and find the page in there. If the pages had been lower in the hierarchy (if they had taken longer to find through browsing) I may have used the history tools more. Answers relate to this context.

Comments Regarding the Visual Design Of SessionGraphs

[*visually intriguing*]

- Again, Netscape just is..... animation is a whole lot more "fun" and easy to understand.
- BI visual design looks cute. I like it.

[*layout needs improvement*]

- Liked it, two points: moving distracting, labels scroll out of the window on the right.
- sometimes they appear too close.
- Though I can move them, it would be better to position them better automatically.

[*additional comments*]

- Netscape only: Only used the personal toolbar.

Comments on "the most positive things about the tool"

[*usefulness*]

- useful.
- I can follow the link in the graph to see a web page just immediately visited before another web site by seeing how near the

- nodes in the graph.
- move directly to sub-pages or upper pages
- to be able to make sessions and show compressed contents below
- very good, especially the large image with title and the concept of session.
- appreciated seeing the path I followed -- it would be interesting to see what happens over a much longer session?
- larger graphs were very useful when I remembered to use them

[*good overview*]

- Speed, ease of access to pages. Overview of browsing pattern (although see next point)
- good overview and some detail

[*visually clear*]

- not very colorful

[*session concept*]

- I liked the task and session concepts and think this would become even more useful overtime
- Session is very useful for most people, but task concept is not.
- It goes back to Ben and Catherine's comments about profiles in class -- one browsing session might relate to one aspect of my work, the next to another -- this way you can keep the histories separate

Comments on “the most negative things about the tool”

[*layout algorithm should be improved*]

- overlapping nodes and links
- Labels sometimes hard to see/read/interpret.

[*page titles should be improved*]

- The abbreviation in the graph is not quite understandable since I don't pay attention to the title of the web pages.
- Sometime it was hard to see the labels or difficult to figure out from the label what the page was (abbreviations, but the bottom window showed details) (frustrating score)

[*manual interaction could be improved*]

- re-shaping graph is not so easy

[*visual concept*]

- The session graphs. Sometimes hard to

read and hard to remember which pages belong to which session.

- Animation effects sometimes make me press wrong place. Node size is a little bit small.
- Sometime hard to interpret the order of pages visited because of the network display.

[*space consuming*]

- always on top (occupying the screen space).
- The browsing icon takes some screen area..

[*additional comments*]

- would be good if I can name the session. in a smaller graphs without title, the session names of each graph might be helpful
- I had a tendency to try to find the right page in the smaller session maps -- forgetting that I could take advantage of the labels in the larger session map.
- Small image without title was not useful to me, even I didn't realize that.

Suggestions for improvement

[*layout algorithm should be improved*]

- Sometimes browsing icons got confused with overlapping nodes and links.
- Overlap of labels, nodes and links confuse me sometimes.

[*page titles should be improved*]

- Maybe showing the full name of each webpage would help users to recognize it more easily.

[*visual concept*]

- could add icon instead of circular nodes
- I liked the visual nature of it, but maybe bringing the overview and detail closer (second and fourth window) would help with interpreting the graphs. The third window (many graphs) is good for navigating between sessions should be first or last.
- The use of thumbnail: liked them, would like them closer to the graphs.
- some information about a session could be given when the mouse is on a session in the smaller graphs window (possibly, the name of the session or the first page added to that session)
- I think it would be nice if the graph is replaced by a tree structure according the

hierarchy of a web site, or other kind of structure that is more intuitive.

[*session concept*]

- I don't know what I need to create a new session in BI
- it may be better if i can integrate more than two sessions into one. but it may make it too large and complicated.
- Opportunity to name sessions and to search for text. Sometimes I forgot in which session I looked at a particular page and was harder to find it. Show the name of teh session on the graph overview windo. (Third window)
- Session identification.
- I cannot feel difference between tasks and sessions. If we use this, I may use only tasks, not sessions because it may not easy to decide whether I need a new session for the same task.
- Every session needs a name, and in the third window, these names can be shown with the sessions. That will make it easier to select the session to use.

[*space consuming*]

- Make the BI an icon that always on top, but the four windows can be hidden. When the user wants to use it, he/she can click it up.
- Would be great to have it with easy open/close funtions for the BI window. (one button on toolbar)

[*additional comments*]

- I wish the font is bigger.
- Some direction information(directed graph)
- Interactive keyword search though the graph. [full text search was not introduced to the subjects]
- changing shape of mouse icon when dragging and dropping
- We may need to have a link for same page in different sessions.
- Drag and drop naming

Sessions & Tasks

- I really like the idea of dividing into tasks and sessions. I would also be interested in seeing what happens over time -- do sessions and task maps become so crowded that they are unusable?
- i think the power of the browsingicon can be shown more easily if the tasks are more difficult and if there are more tasks to more pages and if the pages do not have a link to the home of a park and if the tasks involves deeper hierarchies with more sub-branches.
- I didn't use the ability to move the map around to look at different parts of it -- but could you lose context or find yourself unable to return to the start? -- this might be where the search feature comes in.... not something you can test in an hour.
- Suggestion -- longitudinal studies

9.4.3.2 Second Meeting - After Some Days

Which system did you like more to revisit pages after some days?

[*all preferred SessionGraphs, for these reasons*]

- browsingicon because it's hierarchical and visual structure helps me to remember and locate easily and faster
- BIs: Can see the structure: no need to recall them Can visit the right page with just one click.
- BI was better because I didn't have to switch windows, but liked the Netscape history was better because I could enter a text search. I did not use the shape of the

graphs.

- I like BI because it organizes the web page history very well.
- BI. I can click on the node to visit the place directly. It is a tool delicated to revisit the past web pages.
- BI. More intuitive and gave me visual leads.
- i liked BI better but it is partly because there were many history lists in netscape history.
- BI.Because the history is recorded very clearly, and seperately, so we can find the webpage we want to go very easily.

- I much preferred BI -- the amount of irrelevant and extraneous material that appears overtime in a Netscape history is very distracting.
- BI allowed me to go right back to the session the way I had left it, without irrelevant pages. Plus, it's still "fun".
- BI is better because only with netscape, there was no history which I can reuse. but with BI, I could easily find my history which I used a few days.

Room for additional comments regarding the revisitation of pages

- Netscape history was harder to find. Had too many windows.
- I downgraded slightly for with obstacles, because I had to figure out which session I was interested in -- If I had named the sessions, this would not have been a problem.
- For the same reason I downgraded confusing/clear -- because I had to figure out where to go -- and had no real word cues.
- i didn't use the search function in browsing because the node number is small and it's slow for me to type.

Using SessionGraphs: Compared to the first day: was it harder, easier or equally difficult to revisit pages? Why?

[*easier*]

- easier: no need to recall the structure of the pages
- It was easier than the first day because the interface is easy to use and it doesn't require any memorization.
- It is easier. because I am more familiar with the tool. now I know the meaning of edges.
- BIS/ easier. due to visual leads.
- it's easier to use. I could easily be accustomed to the system and find my history which can be reused.

[*the same*]

- nearly the same because i can remember them easily
- was equally easy. pretty helpful to re-visit web pages quickly and easily.
- the same. Because it looks just like that day. As I said earlier, it recorded the history

seperately.

[*harder*]

- Harder, because I forgot which session/task the pages were in (in BI) and did not remember the graph shapes.
- It was slightly harder the second day because I had to remember the sessions (see comments on previous question.)

Using Netscape: Compared to the first day: was it harder, easier or equally difficult to revisit pages? Why?

[*easier*]

- I did use the history tool today (not before) which made it easier.

[*the same*]

- equally difficult: have to find all structures
- It was equally difficult because its history data is not well organized.

[*harder*]

- It was much harder because all the pages were jumbled together. The orders available through sorting or word search are not intuitive or logically related to the search being conducted.
- harder. i just went there directly from the main home page, which is more straightforward for me to do the tasks
- it is harder. Since I forget what I have added to the toolbar.
- harder. had to recall how to access the root page.
- text-based history list was not very helpful and hard to find the pages I wanted to revisit
- It's harder. Because the history is mixed with the other webpages visited during the last few days, so it's harder to find the one we need.
- Because my old history was deleted there was no advantage to use history information in netscape itself.

Using SessionGraphs: For revisiting pages after some days: what was good about each tool? what was bad?

[*good*]

- help to remember, easy to locate
- Having the thumbnails as reminders of the

- page was good.
- I can recall the set of webpage I visit in the last experiment.
- the relation between pages I have visited were preserved.
- in general, it was helpful.
- It makes the user very easy to find the pages visited several days ago.
- The pattern of my previous search was readily apparent. I was able to see what I had done before and go directly to the appropriate page by either visually scanning the graph or using my mouse to make the abbreviated names clearer.

[*bad*]

- letters are hard to read some times.
- Did not use shape.
- small mouse movement was interpreted as an attempt to move the graph..
- As I've said before -- if I had labeled the sessions it would have been easier to return to the correct one. The ability to name sessions becomes invaluable as the graphs grow larger and more of them are created.

Using Netscape: For revisiting pages after some days: what was good about each tool? what was bad?

- nothing

[*good*]

- Free text search was good.
- It is natural for me to search the webpage again on the web.

[*bad*]

- Good about Netscape's History file? Well all the pages visited were there....
- It doesn't provide sufficient ways to revisit pages after some days.
- However, having to filter out the numerous extraneous pages that were added to the history between related searches caused confusion and slowed the process enor-

- mously.
- history menu is not easy to use
- since after some days some other URLs are added
- even that I must use find menu to look up the right page
- Did not have time to figure out what order they were in the history but looked confusing.
- netscape did not keep pages I visited on the 'back button'.
- history list wasn't very helpful to re-visit pages quickly

Only for SessionGraphs: Did you recognize the shape of your graphs again? What could we do to make them more recognizable?

[*rather yes*]

- quite a bit. no suggestion now
- Yes: if I can annotate (change/modify) the the node names it would be more recognizable.
- Yes, I did recognize them naturally.
- Yes, I recognize it. Now I know that the edge of the graph.
- I can follow the edge to search for the page I have visited before.
- es I did. maybe you want to add names or titles.
- Yes. Labeling sessions on the lower map?
- yes I did recognize the shape of the graph again -- in my case returning to a page causing a cycle made this very obvious.

[*rather no*]

- No. Naming and grouping them would help. (Grouping BIs)
- I cannot remember the graph structure which was generated a few days ago. What I reused is just the page titles.

9.5 Data Related to Longitudinal SessionGraphs Study

This section refers to the longitudinal SessionGraphs (cf. chapter 4.3). It presents didactic information material that was offered to the participants to ease learning.

9.5.1 Online Tutorial and Help

The information subsequently reprinted in the framed field was available to the participants on a web site that accompanied them in the long term study. Since all participants were German, this information was presented in their native language.

It covers the following topics:

- System requirements
- About the study
- What does it mean for you to participate?
- Installation guide
- De-installation
- The first steps using the SessionGraphs tool
- Features of SessionGraphs (all illustrated by text and film, see below)
- The three modes in which the system can operate
- Downloading new versions
- Known Bugs
- FAQ

Studie zu Browsing Icons

Die Installation dauert ca. 10-30 Minuten.

System Voraussetzungen

1. Windows **2000, XP, NT**, manche **98er** Systeme.
2. MS **Internet Explorer 6**.
3. Mindestens **1024 x 768** Pixel Bildschirmauflösung, höhere Auflösung ist empfohlen.
4. **Java 1.3.0 oder höher**.
5. Man braucht **Administrator Rechte**, um die Software zu installieren.

Über die Studie

- Die Hauptaufgabe des Browsing Icons Tools ist es, den Benutzer beim **Wiederbesuch von Webseiten** zu unterstützen - sei dies nach Sekunden, Minuten, Tagen, Wochen oder Monaten.
- **Ziele** dieser Studie sind:
 - das Tool unter "echten" Bedingungen einzusetzen
 - die Stärken des Tools und Verbesserungswürdiges zu lokalisieren
 - die Ergebnisse in anonymisierter Form in meine Dissertation einfließen zu lassen
- **Verfahren:**
 - die Benutzer verwenden das Tool 5 Wochen lang
 - dabei werden all ihre Pfade im Web aufgezeichnet und in Log-Files gespeichert
 - nach ca. einer Woche und am Ende der Studie werden Interviews (je ca. 45 Min.) geführt

Was die Teilnahme bedeutet

- Wer kann mitmachen? Jede/r die/der Lust hat, ein neuartiges visuelles Web-Navigationstool zu benutzen. Im [abstract](#) findet ihr etwas über die Grundidee des Tools. Schaut euch das bitte vor der Studie an. Diese [Demos](#) (Flash-Movies) geben einen Eindruck von der Benutzung.
- Jeder, der teilnimmt, sollte häufig zu Recherchezwecken mit dem Web arbeiten. Ca. 30 Min pro Tag oder länger wäre gut. Mit Recherche ist hier jede relativ zielgerichtete Suche oder Erkundung im Web gemeint.
- In den 5 Wochen sollte man vorwiegend von einem Rechner aus ins Web gehen.
- Wenn das Tool nicht gefällt oder technische Probleme entstehen, kann natürlich jederzeit zum alten Browser ohne Browsing Icons gewechselt werden.
- Die Logfiles sind von Zeit zu Zeit an mich zu schicken. Ich anonymisiere wie gesagt alle Daten vor der Veröffentlichung in meiner Dissertation. Dort tauchen nur statistische Werte auf und ausgewählte Beispiele, die mit euch besprochen wurden.
- Das Tool ist kein fertiges Produkt, sondern ein Prototyp, der einige Grundfunktionalitäten implementiert. Manche Funktionalitäten sind noch nicht umgesetzt und an einigen Stellen treten auch noch Fehler auf. Erwartet bei der Benutzung also kein glatt laufendes System.
- Meldet euch gerne mit Fragen oder bei Problemen: mayer@informatik.uni-hamburg.de oder Tel.: 040 42883 2356 oder 0177 4800 691.

Installations Anleitung für die erste Installation :

Die erste Installation des Tools benötigt 2 bis 4 Schritte. Schritt 1+2 sind notwendig, 3+4 je nach Bedarf.

Schritt 1: "Browsing Icons" installieren

Schritt 2: "IESpy" installieren (dient der Kommunikation zw. Java und Br. Icons).

Schritt 3: ggf. Java installieren

Schritt 4: ggf. MS Internet Explorer 6 installieren

1. Installation von Browsing Icons:

- Die Browsing Icons Software visualisiert die Web-History und macht Wiederbesuche möglich. Sie speichert die persönliche History auf der Festplatte.
- Falls du die Installations-CD hast, öffne das Verzeichnis "**1_install_browsing_icons**".
Sonst lade es hier herunter und entpacke es auf deine Festplatte:
[zur aktuellen Version](#)
- Starte "**setup.exe**".
- Wichtig: Während der Installation kann die Nachricht: "**CRC Errorcab file ... contact your software vendor**" erscheinen. **Ignoriere sie einfach** (klick "ignore")!
- Die Software kann in ein beliebiges Verzeichnis gespeichert werden.
- Unter "start->programme->Browsing Icons" findet ihr einige Einträge, um das Programm zu starten und einzurichten.

2. Installation von IESpy:

- IESpy ist ein Hilfsprogramm von Hartmut Obendorf and Torsten Hass, das die Kommunikation von Java und MS Internet Explorer (MSIE) erlaubt.
- Falls du die Installations-CD hast, öffne das Verzeichnis "**2_install_iespy**".
Sonst lade es hier herunter und entpacke es auf deine Festplatte:
[2_install_iespy.zip](#) (ca. 1 MB).

Ein Installer für **Win98** (läuft auf den meisten 98er Systemen):
[IESpy_2003_01_15_win98.zip](#) (ca 1 MB)
- Starte "**SETUP.EXE**".
- Die Software kann in ein beliebiges Verzeichnis gespeichert werden. Merke dir das Verzeichnis für den ersten Start Dort muss der Pfad noch einmal eingetragen werden. (default: C:\Programme\ObendorfandHass\IESpy).
- IESpy wird automatisch von Browsing Icons gestartet. Dabei wird leider ein "unnötiges" Fenster geöffnet, das Platz in der Windows-Bar belegt aber geöffnet bleiben muss. Wenn ihr das umgehen wollt, könnt ihr IESpy einmal am Tag von Hand starten (und abends schliessen). Dazu gibt es unter start->programme "IESpy" 2 Einträge. In Browsing Icons "Settings" muss dazu noch die Checkbox "start & stop IESpy automatically" unmarkiert sein.

3. Java Installation:

- Wenn ihr nicht sicher seid, ob Java bereits auf eurem System installiert ist, fahrt einfach mit Schritt 4 fort. Bei Problemen solltet ihr dann ggf. Java nachinstallieren.

- Wenn es auf dem System liegt, aber nicht läuft, stellt sicher, dass die PATH variable richtig gesetzt ist. Siehe <http://ccism.pc.athabascau.ca/html/vhd/javapath.xml>.
- Ihr koennt auch Java neu installieren:
- Auf der CD findet ihr den Installer für Java 1.4.1 in **3__OPTIONAL__install__java__1.4.1__o1**. Das JAVA 1.4.1 **JRE** (Runtime Environment) ermöglicht es, Java Programme laufen zu lassen, das **SDK** 1.4.1 (Software Development Kit) erlaubt auch die eigene Programmierung. Installiert eines von beiden.
- Online download: <http://java.sun.com/downloads/>

4. MS Internet Explorer 6:

- Ihr benötigt MSIE 6.0 oder höher.
- Auf der CD findet ihr einen kompletten Installer (und einen Web-Installer) unter **"4__OPTIONAL__install__ms__internet__explorer__6"**.
Online gibt es die neueste Version unter: <http://www.microsoft.com/windows/ie/default.asp>


5. Einmaliges Einrichten des MS Internet Explorer 6:

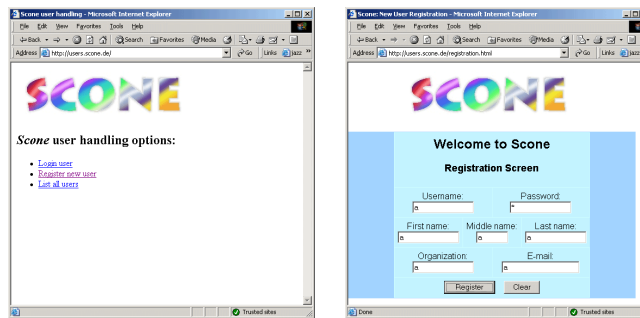
- Um Browsing Icons zu benutzen (mit oder ohne die Visualisierung, siehe 3 Modi unten), muss der Internet Explorer speziell eingestellt werden. Ruft dazu folgenden Eintrag auf:
start->Programme->Browsing Icons->Prepare IE for using Br Icons (once).
Das müsst ihr nur einmal am Anfang der Studie machen.
Für technisch Interessierte: Hier wird die Windows registry wie folgt geändert: "localhost:8088" wird als proxy server eingetragen, "*.scone.de" als trusted site. IE wird so eingestellt, dass er bei jedem Seitenbesuch nach einer neuen Version der Seite schaut, und IE wird erlaubt, den Fokus abzugeben.
- Um IE wieder einmal ganz ohne Browsing Icons und ohne Logging benutzen zu können:
start->Programme->Browsing Icons->Reset IE for normal use (once).
Benutzt das auch, wenn ihr irgendwelche Probleme mit Browsing Icons habt und IE wieder normal benutzen wollt.
- Internet Explorer muss der **primäre Web browser** auf dem System sein. Das erreicht ihr wie folgt:
Win XP: es gibt einen speziellen Eintrag im "Start" menu.
Andere Plattformen: In "Internet Options"->"Programs" markiert "IE should check to see whether it is the default browser". Startet IE neu. Dann solltet ihr gefragt werden, ob IE der primäre Browser sein soll.
- Erlaubt die Benutzung von **Cookies**. ("Privacy" tab in Internet Options auf "medium" oder weniger).

De-Installation:

- Ihr könnt Browsing Icons und IESpy einfach über Systemsteuerung-Software deinstallieren.
- Davor(!) klickt einmal auf **start->Programme->Browsing Icons->Reset IE for normal use (once)**. Dadurch wird die Windows Registry zurückgesetzt und IE läuft wieder normal.

Die ersten Schritte mit Browsing Icons - Tipps zum Einstieg

1. Startet Browsing Icons per Start->Browsing Icons->”Start Browsing Icons with visualization and with logging” .
2. Wenn das Tool nicht startet, überprüft, ob Java richtig installiert ist. Öffnet eine DOS-Shell und gebt ”java -version” ein. Diese muss mindestens 1.3.0 sein. Dann seht ihr, ob Java richtig installiert ist.
3. Falls ihr beim ersten Start zur Scone Homepage geleitet werden, gebt einfach irgendwelche Daten in die Felder ein, z.B. ”a”, ”a”, ”a” ... Diese sind für die Studie nicht wichtig.



4. Glückwunsch ;-) Das meiste ist geschafft!!!
5. Geht in das Browsing Icons ”settings” Menü und öffnet ”configure settings”. Tragt in den vier Feldern folgendes ein: den URL eurer **Liebblings-Suchmaschine** (diese ist dann über den Button mit der Lupe zu erreichen). Eine **Startseite**, die immer zu Beginn geladen wird. Euren **Namen** und eure **Email-Adresse**. Schliesst das Fenster und speichert die Änderungen, wenn ihr gefragt werdet. Tragt hier auch das **Installationsverzeichnis von IESpy** ein, sofern es nicht das bereits eingetragene unter C: ist. Ohne diesen Eintrag funktioniert das Tool nicht.

Features von Browsing Icons

Um einen Eindruck von der Benutzung zu bekommen, könnt ihr euch die [Demo Filmchen](#) ansehen. Dort werden die wichtigsten Benutzungsszenarien einmal durchgespielt.

In order to fully use its potential be sure to understand also the special functionality described on there. Try to use all the functions in the beginning in order to see how they work and feel. Probiert zu Beginn einmal alle Funktionen aus, die es gibt, damit ihr sie später nach Bedarf einsetzen könnt:



- Eine neue **Session** anlegen (per Menü / Button): Die enthält dann die Webseiten
- Eine neue **Task** (Aufgabe) anlegen, um Sessions hierarchisch zu organisieren.
- Besucht ein paar Webseiten, um zu sehen wie das Tool reagiert.
- Rechts oben mit dem **on/off** Button könnt ihr zwischendurch das Verfolgen neuer Besuche in der Visualisierung unterbinden. Diese Seiten können dann aber auch nicht per Visualisierung wiederbesucht werden. Bekommt ein Gefühl, wann es sinnvoll ist, das zu tun.
- Versucht per rechtem Mausklick aus: **Mail Session to a friend** (wenn ihr mögt an mich).
- Geht in 2 Sessions mal den selben Weg lang. Dann erscheinen oben sog. **Browse Ahead Graphs**, blau gefärbt. Dadurch habt ihr die Möglichkeit, gleich zu bestimmten "Ziel"-Seiten zu springen, ohne den ganzen Pfad dorthin "zu Fuß" zurückzulegen.
- Probiert den **Lupe**-Button aus zur Suche im Web.
- Button links neben der Lupe: eine Alternative zum "**Back**"-Button - er löscht die Darstellung der letztbesuchten Seite. Gut bei versehentlichen Besuchen.
- Mit **Shift-Click** könnt ihr Knoten in der Visualisierung selektieren. Mit Shift-Drag ganze Bereiche (mit Strg ist die Selektion erweiterbar). So könnt ihr einzelne Knoten auswählen und z.B. löschen.
- Die beiden **Fernglas** Buttons dienen der Textsuche in der History (durchsucht Titel und URLs, nicht den Seiteninhalt). Zeigen nacheinander alle Seiten in Browsing Icons rosa gefärbt an.
- Versucht mal, per **drag and drop** im obersten View eine Session in eine andere Task zu ziehen. Bzw. sie mit Strg-drag dorthin zu **kopieren**.
- Kuckt, was das rechte Mausmenü auf den Knoten im Graphen bietet: **färben**, **mailen**, den View wieder "**verflüssigen**", etc.
- Browsing Icons speichert den Zustand nach grösseren Änderungen der History. Im Edit Menü könnt ihr zu früheren Versionen zurück.

Eure Kommentare und Bug-Reports: Ganz wichtig sind eure individuellen Kommentare zum Tool. In den Momenten, in denen ihr denkt: Wow, diese Eigenschaft finde ich gut, oder in dieser Situation bringt's mir viel, bitte ich euch, einen kurzen **Kommentar abzuspeichern** (Menu "Extras->save a comment"). Dieser wird dann ins Logfile geschrieben und ermöglicht mir, euch in den Interviews gezielt danach zu befragen. Gebt auch bitte Kommentare ab zu Aspekten, die verbessert werden sollten. Speichert diese auch über diesen Weg im Logfile. Bei echten Bugs, d.h. Fehlfunktionen, die ich möglichst noch während der Studie beheben soll, bitte ich euch, [mir eine kurze Mail zu schicken oder mich kurz anzurufen](#).

Starten und Beenden des Tools

Starten: Wenn ihr Browsing Icons startet, startet IE automatisch. Ihr braucht IE nicht mehr direkt zu starten. Das Tool kann in 3 Modi gestartet werden. S.u.

Benutzt zum Starten nur die beiden Shortcuts im Start-Programme Menü. Die könnt ihr per Strg-Click auch auf den Desktop oder in die Fußleiste von Windows ziehen.

"Start Browsing Icons with Visualization and with Logging"  (für ICONS-ON or -OFF-MODE, s.u.) "Start Browsing Icons without Visualization but with Logging"  (für PLAIN EXPLORER-MODE).

Beenden: Menü Sessions->Exit oder Strg-Q oder Klick in rechte obere Ecke. Nicht durch Beenden der zugehörigen DOS-Shell, sonst wird euer Pfad nicht gespeichert. Im Modus, wo das Tool selbst nicht sichtbar ist, koennt ihr es mit rechtem Mausklick auf das Icon in der Windowsbar beenden.

Während des Laufs und beim Beenden kommt es zu einigen **Fehlerausgaben** in der zugehörigen DOS Shell. Diese sind gewöhnlicherweise bei sonst normalem Systemverhalten nicht weiter von Bedeutung.

Etwas zur Stabilität

Es kann passieren, dass das Tool aus nicht definierbaren Gründen abbricht, oder beim Laden die History plötzlich leer erscheint, obwohl man schon lange gesurft ist. Hierbei können eure History Daten verloren gehen. Ich habe diverse Backup Mechanismen eingebaut, so dass in vielen Fällen noch auf eine ältere Version zurückgegriffen werden kann. Wie ihr diese laden könnt, sage ich euch dann im individuellen Fall. Meldet euch in solchen Fällen gerne gleich.

Die 3 verschiedenen Modi des Tools

ICONS-ON-MODE

Der normale Modus für die Studie:

- Das Tool ist rechts sichtbar
- Euer Pfad wird visualisiert
- Das Logfile zur späteren Analyse wird geschrieben
Versucht, so oft wie möglich in diesem Modus zu surfen.

Um in diesem Modus zu surfen:

 Start->Programme->Browsing Icons->Start B I. with visualization and with logging.

Wenn der on/off Button rechts oben grün umrandet ist ("on" fett) dann läuft das Tool in diesem Modus. Wenn "off" dann clickt einmal auf diesen Button.

ICONS-OFF-MODE

Der Modus um zwischendurch den Pfad nicht zu visualisieren:

- Das Tool ist rechts sichtbar
- Euer Pfad wird NICHT visualisiert
- Das Logfile zur späteren Analyse wird geschrieben

Ihr könnt das Tool währenddessen trotzdem zum Revisit alter Seiten benutzen. Seiten, die ihr in dem Modus neu besucht, können per Tool nicht wiederbesucht werden.

Benutzt diesen Modus, wenn ihr sicher seid, dass euch eine dauerhafte Visualisierung des momentanen Pfads nicht interessiert. Ihr haltet so die Anzahl der Sessions und Seiten in Grenzen.

Um in diesem Modus zu surfen:

 Start->Programme->Browsing Icons->Start B I. with visualization and with logging.

Wenn der on/off Button rechts oben rot umrandet ist ("off" fett) dann läuft das Tool in diesem Modus. Wenn "on" dann clickt einmal auf diesen Button.



Visual history on: - Visual history off:

PLAIN EXPLORER-MODE

Ganz ohne Visualisierung aber trotzdem mit Logging:

- Das Tool ist rechts NICHT sichtbar
- Euer Pfad wird NICHT visualisiert
- Das Logfile zur späteren Analyse wird geschrieben
- Ihr könnt das Tool währenddessen NICHT benutzen

Benutzt diesen Modus, wenn ihr IE ohne Visualisierung benutzen wollt, aber dennoch die für die Studie relevanten Logfiles produzieren wollt.

Um in diesem Modus zu surfen:

 Start->Programme->Browsing Icons->Start B I. without visualization but with logging.

MS Internet Explorer pur oder euer Browser der Wahl

Nur Internet Explorer

Natürlich könnt ihr auch immer wieder ganz auf Browsing Icons __und__ Logging verzichten. Dadurch fallen dann aber die für die Studie wichtigen Daten nicht an. Es kann trotzdem sinnvoll sein, das zu tun, wenn z.b. technische Probleme auftreten.

- Das Tool ist rechts NICHT sichtbar
- Euer Pfad wird NICHT visualisiert
- Das Logfile zur späteren Analyse wird NICHT geschrieben
- Ihr könnt das Tool NICHT benutzen

Um dies zu ermöglichen wählt folgenden Eintrag (setzt die IE Registry Einträge zurück). Start->Programme->Browsing Icons->Reset IE for normal use (once). Oder direkt im Installationsverzeichnis von Browsing Icons: config_iexplorer_NoProxy_NoBrowsingicons.reg.

Download neuer Versionen

Hier findet ihr ggf. Updates zum download. Das Nachinstallieren ist einfach und wird hier jeweils beschrieben.

Oben = aktuelle Version	Beschreibung / Unterschied zur Vorversion
<p>Version 0.9.5 komplett: browsingiconso.9.5_inst.zip (7,5MB)</p> <p>Upgrade von 0.9.3 oder 0.9.4 auf 0.9.5: biv.jar (500K - ersetzt die gleichnamige Datei in "browsingicons/mmlib") htmltokenizer.jar (30K - ersetzt die gleichnamige Datei in "browsingicons/lib")</p> <p>zusätzlich für Upgrade von 0.9.3 auf 0.9.5 (nicht nötig wenn 0.9.4 vorhanden) browsingicons.xml (1K - speichern in "browsingicons/config/scone")</p>	<ul style="list-style-type: none"> • Wenn ihr eine Homepage ("StartURL" in BI-Settings) angebt, die von der voreingestellten abweicht, erscheint nicht mehr nach jedem Hochfahren eine neue Session • Wenn nach dem Hochfahren von BI mal wieder alle früheren Sessions gelöscht scheinen: jetzt kein umständliches Zurücksetzen des Zählers in der Textdatei mehr, sondern einfach Menü-Edit-"Go back to previous version" sooft klicken, bis die Sessions wieder da sind. Sollte selbst das nicht gehen, bitte kurz Mail an mich, ich sag euch dann, was ihr noch tun könnt. • Die Internet Movie Database verursacht keinen Fehler mehr (Dank für die Scone-Überarbeitung an Harald Weinreich!). • www.spiegel.de lädt ordentlich (ebenfalls dank an Harald!).
<p>Version 0.9.4 komplett: browsingicons.zip (7,5MB)</p> <p>upgrade von 0.9.3 auf 0.9.4: biv.jar (500K - speichern in "browsingicons/mmlib") browsingicons.xml (1K - speichern in "browsingicons/config/scone")</p>	<ul style="list-style-type: none"> • bei Problemen oder Fehlern kann ein Error-Log-File generiert werden. Einfachere Fehlersuche für mich. <p>Um ein Errorlog zu generieren, öffnet die Datei <code>browsingicons/config/scone/browsingicons.xml</code> in einem Texteditor. Setzt "Generate Error Output File" auf "true". Bei nächsten Lauf von Browsing Icons wird die Datei <code>"browsingicons/run/errors.txt"</code> ein ausführliches Ablaufprotokoll enthalten. Schickt mir bei Problemen bitte diese Datei.</p> <ul style="list-style-type: none"> • ein kleiner Bug weniger (ein Eintrag aus den Settings wurde nicht übernommen).
<p>Version 0.9.3 1 install browsing icons 0.9.3.zip (7,5 MB).</p>	

Verzeichnis bekannter Bugs

Hier findet ihr ein Verzeichnis der derzeit bekannten Bugs. Und ggf. die Versionsnummer, ab der sie behoben wurden. Hier findet ihr die [Versionen](#).

Titel	Beschreibung	fixed? (seit Version)
Einsame Kugel	Manchesmal erscheint eine Kugel, meist die 3. in einer neuen Session, ohne Verbindung zur letzten und nächsten Kugel.	
Fenster schliessen	Es schliessen sich beim Runterfahren von BI leider nicht alle MSIE Fenster automatisch. Das ist ein Bug in MSIE.	
Ebay Fehler	Ebay Seiten werden nicht richtig repräsentiert.	behooben, vo.9.2
Window Bar oben oder unten	Seit vo.9.2 kann man per settings die Position der Windowsbar angeben.	behooben, vo.9.2
Verlorene Sessions	Manche Sessions werden nach neu laden im Task View nicht angezeigt.	behooben, vo.9.2
Tasks und Sessions: nicht umbenennbar.	Bug in Java 1.4: Sessions and Tasks können nicht umbenannt werden. Seit v o.9.1 geht das per rechtem Mausklick auf den Task / Sessionknoten.	behooben, vo.9.1
Titeländerung	Titel von Seiten werden unvermittelt bei neuem Lauf geändert.	behooben, vo.9.2
Absturz wegen Tasktiefe	B. Icons speichert die Task-Hierarchie in einer Datei-Hierarchie. Wenn die Pfade länger als 255 Zeichen werden (unter Windows nicht erlaubt), bricht B. Icons unvermittelt ohne zu speichern ab.	-
Fehlende Verbindungen	Manchmal werden Knoten ohne Verbindung zu anderen Knoten zugefügt.	-
HTML-Frame Problem	Bei einigen Frameseiten kommt es zu Problemen	-

FAQs

[Hier](#) findet ihr Fragen und Antworten, die euch vielleicht helfen.

9.5.2 Tutorial Movies

The following information was provided on a web page that included a step by step video tutorial to get familiar with the tool. See:

asi-www.informatik.uni-hamburg.de/personen/mayer/phd/examplesofuse.html

A. Compact Introduction (all movies in only three separate files)

If you watch these 3 movies, you know most about Browsing Icons and how to use it. Each movie is about 10 minutes long.

Some people discovered **problems** playing these flash movies. Try the smaller ones below (same content, just in single files). Or contact me so I can send you a CD or assist you personally.

- Watch the Growth of an Example-Session
- Main functionality - How web pages can be revisited using Browsing Icons
- Additional functionality - Explaining the Menus

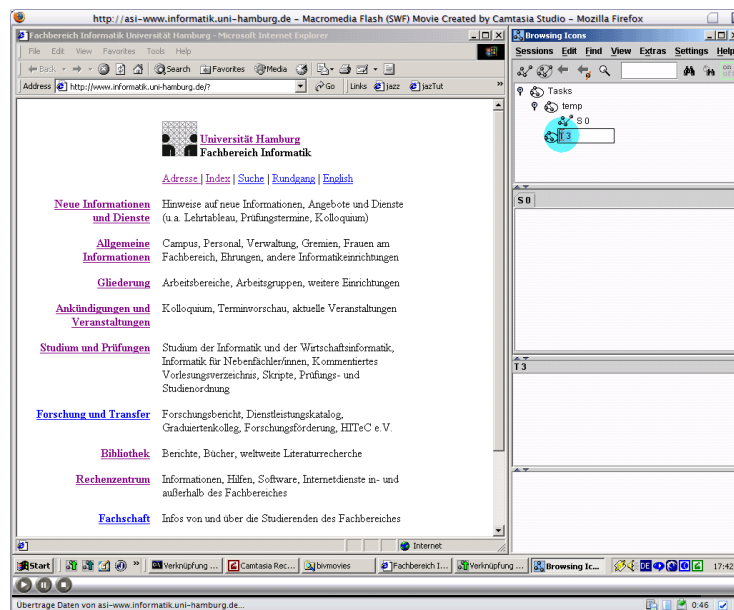


Figure 9-7: Tutorial movie 1 - introduction to SessionGraphs.

B. Single Movie Files for specific Topics

Watch the Growth of an Example-Session

All 4 Movies together (8'47") Flash - online | AVI - download or local

1. Introducing the scenario and creating 3 Tasks. (1'36")
2. Starting the first Session. (2'00")
3. Growing the Session. (2'39")
4. Half an hour later - many Sessions and the first Revisits. (2'30")

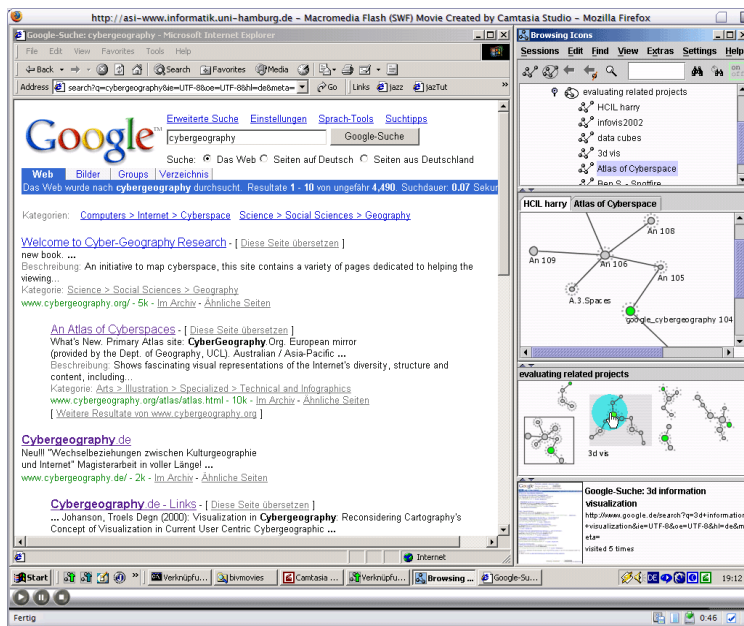


Figure 9-8: Tutorial movie 2 - an exemplary session.

Main functionality

How web pages can be revisited using Browsing Icons

All 7 Movies together (9'12")

1. Revisits within a Session. (2'52")
2. Revisits within a Task. (1'10")
3. Revisits within the whole History. (0'42")
4. Multi-Page Revisits. (0'56")
5. Reusing Tasks: similar and repeated Tasks. (1'12")
6. Identifying identical paths: Look-Ahead-Graphs. (1'19")
7. Sharing Histories in a Group: E-Mailing Sessions to your Friends. (1'19")

Additional functionality

Explaining the Menu

All 10 Movies together (11'26") Flash - online | AVI - download or local

1. The Sessions Menu. (1'43")
2. The Edit Menu. (0'50")
3. The Find Menu. (1'00")
4. The View Menu. (1'05")
5. The Extras Menu. (1'37")
6. The Settings Menu. (2'33")
7. The Help Menu. (0'20")
8. The Back Buttons. (0'32")
9. The On/Off Switch. (1'21")
10. Closing the Application. (0'23")

9.5.3 Consent Form

Each participant signed the following consent form prior to participation:

- I am participating voluntarily in this study.
- I was informed about the duration, proceedings, and my tasks in advance.
- I was able to ask questions before and during the study. These questions were answered satisfactorily.
- I am conscious of being able to discontinue this study at any time, without an explanation. This will not be disadvantageous for me in any way.
- I agree that the data collected during the study (interviews and log files) will be used anonymously in the dissertation of Matthias Mayer.
- I agree to have the interviews audio-taped. Citations will be used anonymously in the dissertation of M. Mayer, if at all.
- My signature affirms all the above points.

9.5.4 Questionnaire for First Interview

In the following, the questions are presented that the subjects were asked in the first oral interview, briefly after the beginning of the study. The answers were recorded on tape, and manual notes were taken during the interviews. The form was manually filled out by the interviewer (me) synchronously to the oral interview. Also here, *BI* or *Browsing Icons* is the old name of SessionGraphs.

Fragen zu Beginn.

Wie läuft BI denn im Moment?

Informelle Sammlung der wichtigsten Fehler.

Allgemeine Angaben zum Umgang mit dem WWW, ohne Berücksichtigung von Browsing Icons.

- 1) Wie würdest du deine Erfahrung im Umgang mit Computern bezeichnen?
Nicht erfahren 1○ 2○ 3○ 4○ 5○ 6○ 7○ 8○ 9○ *Sehr erfahren*
- 2) Seit wievielen Jahren benutzt du Computer?
- 3) Wie würdest du deine Erfahrung im Umgang mit dem WWW bezeichnen?
Nicht erfahren 1○ 2○ 3○ 4○ 5○ 6○ 7○ 8○ 9○ *Sehr erfahren*
- 4) Seit wievielen Jahren arbeitest du mit dem WWW? ____
- 5) Wieviele Stunden arbeitest du pro Woche schätzungsweise mit dem WWW? ____
- 6) Welchen Web-Browser verwendest du? ____
- 7) Auf welchem Betriebssystem oder welchen Systemen? ____

Fragen zu deiner bisherigen Art und Weise (ohne Browsing Icons), Webseiten wiederzubesuchen:

- 8) Benutzst du Bookmarks/ Favoriten? ____
- 9) Kannst du grob abschätzen, wieviele Bookmarks du etwa hast? ____
- 10) Benutzst du sie täglich, wöchentlich, monatlich? ____
- 11) Ordnest du sie hierarchisch? ____
- 12) Wann organisierst du deine Bookmarks?
- Nie
 - Vorwiegend sofort
 - Vorwiegend am Ende einer Session
 - Regelmässig, nämlich: ____
 - Unregelmässig
- Sonstiges: ____
- 13) Benutzst du selbst definierte Toolbar Buttons (feste Links zu anderen Webseiten)? / ordnest du diese in eigenen Ordnern? ____
- 14) Zu welchen Tätigkeiten benutzst du das WWW am meisten? ____
- 15) Wie besuchst du Webseiten wieder, die du erst vor ein paar Sekunden oder Minuten besucht hast? ____
- Wichtigkeit folgender Hilfsmittel: 1 = keine, 9 = sehr grosse Bedeutung
- ____ Back Button
- ____ Browser-History
- ____ Benutzerdefinierte Toolbar Buttons
- ____ Bookmarks/Favorites
- ____ URL-Eingabe
- ____ URL-Popup
- ____ Suchmaschine
- ____ Lokale Kopie
- Kommentar ____
- 16) Wie besuchst du Webseiten wieder, die du vor einigen Tagen, Wochen oder Monaten zuletzt besucht hast?
- Wichtigkeit folgender Hilfsmittel: 1 = keine, 9 = sehr grosse Bedeutung
[as above (15)]

Browsing Icons – Du hast ja angefangen, Browsing Icons zu benutzen, ...

- 17) Wie ist dein erster Eindruck? ____
- 18) Was sind die wichtigsten Punkte, die am System verbessert werden sollten?
Weitere Kommentare und Feedback zu diesem Interview oder zur Studie?

9.5.5 Questionnaire for Second Interview

The following questions were presented to the subjects in the second oral interview, at the end of the study. The answers were recorded on tape, and manual notes were taken during the interviews.

Allgemeine Angaben zum Umgang mit Browsing Icons

- 1) Wie würdest du deine Erfahrung im Umgang mit Browsing Icons bezeichnen?
Nicht erfahren 10 20 30 40 50 60 70 80 90 Sehr erfahren
- 2) Wieviele Stunden hast du pro Woche schätzungsweise mit BI gearbeitet? ____

Fragen zu deiner Art und Weise MIT Browsing Icons Webseiten wiederzubesuchen

- 3) Zu welchen Tätigkeiten hast du während der Studie das Tool am meisten benutzt?

- 4) Wie besuchst du (mit Browsing Icons) Webseiten wieder, die du erst vor ein paar Sekunden oder Minuten besucht hast?

Wichtigkeit folgender Hilfsmittel: 1 = keine, 9 = sehr grosse Bedeutung

- ____ BI Sessionview
- ____ BI Taskview
- ____ BI Back-Button
- ____ IE Back Button
- ____ Browser-History
- ____ Benutzerdefinierte Toolbar Buttons
- ____ Bookmarks/Favorites
- ____ URL-Eingabe
- ____ URL-Popup
- ____ Suchmaschine
- ____ Lokale Kopie

Kommentar

- 5) Wie besuchst du Webseiten wieder, die du vor einigen Tagen, Wochen oder Monaten zuletzt besucht hast (mit BI)? ____

Wichtigkeit folgender Hilfsmittel: 1 = keine, 9 = sehr grosse Bedeutung

[as above for short-term revisits]

Du hast ja angefangen, Browsing Icons zu benutzen, ...

- 6) Wie ist dein derzeitiger Eindruck vom System? ____
- 7) Was sind die wichtigsten Punkte, die am System verbessert werden sollten? ____
- 8) Was sind die wichtigsten Punkte, die am Konzept verbessert werden sollten (im Gegensatz zur Implementation - es macht nichts, wenn du dich schon bei der letzten Frage darauf bezogen hast)? ____
- 9) Was gefällt dir am meisten am System (mehrere Angaben möglich)? ____
- 10) Was gefällt dir am meisten am Konzept (falls noch nicht beantwortet)? ____

- 11) Jetzt vergleichen wir deinen bisherigen Web-Browser und Browsing Icons in ihrer Fähigkeit, dich beim Wiederbesuchen von Webseiten zu unterstützen. Versuche dabei das Konzept von Browsing Icons zu bewerten und von derzeitigen Bugs etc. abzusehen.: _____
Welchen Web-Browser bewertest du? _____
- 12) Wie gut unterstützt er dich bei Wiederbesuchen nach einigen Sekunden oder Minuten (ohne Browsing Icons)?
schlecht 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *gut*
- Wie gut unterstützt er dich bei Wiederbesuchen nach einigen Tagen, Wochen oder Monaten (ohne Browsing Icons)?
schlecht 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *gut*
- Wie gut unterstützt dich Browsing Icons bei Wiederbesuchen nach einigen Sekunden oder Minuten
schlecht 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *gut*
- Wie gut unterstützt dich Browsing Icons bei Wiederbesuchen nach einigen Tagen, Wochen oder Monaten?
schlecht 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *gut*
- Kommentar. _____

Jetzt zu einigen Details von Browsing Icons:

- 13) DER SESSIONVIEW
Das visuelle Layout von Sessions: die räumliche Anordnung von Knoten und Kanten
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
nach mehreren Tagen
nicht wiedererkennbar 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *gut wiedererkennbar*
- 14) Die visuelle Repräsentation einzelner Webseiten als Kreise
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 15) Die Animation neuer Knoten (Einfügen, Bewegen, Ausgleichen)
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 16) Die manuelle Umformbarkeit des Graphen
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 17) Das langsame "Einfrieren" des Graphen
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 18) Die Thumbnails (verkleinerte Abbildungen der Webseiten)
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 19) DER TASKVIEW (die verkleinerte Ansicht der Session-Graphen einer ganzen Task)
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 20) DER TASKCHOOSER (die hierarchische Anordnung der Sessions in einer Task-Hierarchie)
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*
- 21) KONZEPTE: Sessions (einzelne Einheiten von Web-Aufenthalten)
nicht hilfreich 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 *hilfreich*

- 22) Tasks (Aufgaben, zu denen einzelne Sessions gehören, hierarchisch geordnet)
nicht hilfreich 10 20 30 40 50 60 70 80 90 *hilfreich*
- 23) Wie gerne würdest du ein solches System auch weiterhin benutzen, wenn es die technischen Mängel, die es derzeit hat, nicht mehr hätte?
nicht gerne 10 20 30 40 50 60 70 80 90 *sehr gerne*

BESONDERE FUNKTIONALITÄTEN

- 24) Drag-and-Drop im Taskchooser
nicht hilfreich 10 20 30 40 50 60 70 80 90 *hilfreich*
- 25) Mail to a friend
nicht hilfreich 10 20 30 40 50 60 70 80 90 *hilfreich*
- 26) Browse Ahead Graphs
nicht hilfreich 10 20 30 40 50 60 70 80 90 *hilfreich*
- 27) undo / redo
nicht hilfreich 10 20 30 40 50 60 70 80 90 *hilfreich*

Zu besonderen Situationen, in denen dir Browsing Icons geholfen hat:

Bitte versuche, dich an eine kürzlich erlebte Situation zu erinnern, in welcher du das Browsing Icons Tool zum Auffinden von Informationen benutzt hast und in der es für dich sehr wichtig war, eine bestimmte Webseite oder mehrere Webseiten wieder zu besuchen, um eine für dich bedeutsame Aufgabe zu erledigen - eine Situation, in der das Tool für dich hilfreich war.

Bitte beschreibe die Situation und dein Vorgehen so detailliert, dass ich mir davon ein klares Bild machen kann. Du kannst auch deine Browsing Icons hier am Schirm verwenden, um es mir deutlich zu machen.

Ggf. wird wie folgt nachgefragt: Was war dein Ziel? Wieso war es wichtig für dich? Wie bist du vorgegangen? Weisst du noch, wann das war? Und wie lange das etwa gedauert hat? Welche Eigenschaften des Tools haben dir dabei speziell geholfen? Wie hättest du dabei noch besser unterstützt werden können? Diese Frage wird für 1-3 Situationen gestellt. Nachträglich wird die Situation im Logfile identifiziert.

Weitere Kommentare und Feedback zu diesem Interview oder zur Studie? _____

9.5.6 Logged Event Types

The following comprehensive list shows, which events have been logged automatically, both referring to the SessionGraphs tool and to the web browser.

SessionGraphs Related Events

System or User Initiated

NEW_SESSION
 NEW_TASK
 NEW_VISIT_IN_SESSION
 USER_CHANGED_ANIMATION
 USER_CHANGED_CONFIGFILE

USER_CLICKED_SESSIONVIEW_TABS
 USER_COLORED_SESSIONNODE

User Interaction Initiated

BIV_BACK_WITH_DELETE
 BIV_BACK_WITHOUT_DELETE
 BROWSINGICON_CLICKED
 END_DRAG_DND_MOVED_NOT_COPIED

END_DROP_DND_SESSIONNODE_IN_TCH
 FIND_NEXT
 FIND
 HOVERED_OVER_TV_NODE
 LOADED_BY_BROWSINGGRAPH
 MOUSECLICK_IN_BAHEAD_ICON
 SEARCH_THE_WEB
 SESSION_DELETED
 SESSION EMAILED
 SESSION_RENAMED
 SESSIONNODE_DELETED_BY_MENU
 SESSIONNODE_DELETED_BY_POPUP
 SESSIONNODE_RENAMED
 START_DRAG_DND_SESSIONNODE_IN_TCH
 SUBMITTET_COMMENT_BEGIN
 SUBMITTET_COMMENT_END
 SUBMITTET_EMAILCOMMENT_BEGIN
 SUBMITTET_EMAILCOMMENT_END
 SYSTEM_SHUTDOWN
 SYSTEM_STARTING_UP
 TASK_DELETED
 TASK_RENAMED
 TREE_EXPANDED_OR_COLLAPSED
 USER_CHECKED_FOR EMAILED_SESSIONS
 USER_CREATED_NEW_SESSION
 USER_CREATED_NEW_SESSION_BY_MAINMENU_O
 R_SHORTCUT
 USER_CREATED_NEW_SESSION_WITH_PAGE_BY_
 POPUP

USER_CREATED_NEW_SESSION_BY_TOOLBAR
 USER_CREATED_NEW_TASK
 USER_CREATED_NEW_TASK_BY_MAINMENU
 USER_CREATED_NEW_TASK_BY_TOOLBAR
 USER_MOVED_BROWSINGICON
 USER_MOVED_SESSIONNODE
 USER_SELECTED_SESSION_BY_TCH
 USER_SELECTED_TASK_BY_TCH
 USER_SWITCHED_BICON_ONOROFF

Microsoft Internet Explorer Related Events

User Interaction Initiated

LOADED_BY_HYPERLINK
 LOADED_BY_HISTLIST_OR_FAVRTES
 LOADED_BY_HOMEBUTTON
 LOADED_BY_FORMSUBMIT
 LOADED_BY_LINKBARSHORTCUT
 LOADED_BY_MSIE_BACK
 LOADED_BY_MSIE_FORWARD
 LOADED_BY_URL_ENTRY_OR_URLPOPUP
 RELOADED_BY_TOOLBAR

9.6 Data Related to Collaborative Web Usage Study

This appendix refers to the web usage study analyzing the behavior of 25 participants using an unmodified web browser (chapter 5).

9.6.1 Individual Contributions of the four Authors

Table 9-2 shows the individual contributions of the four participating authors to the web usage study. This matrix considers only the work necessary for the results mentioned in this thesis, not the work of the entire study.

	Harald Weinreich	Eelco Herder	Hartmut Obendorf	Matthias Mayer
Conceptual Study Design	x	x	x	x
Recruiting & Interviews	x	x	x	x
Data Preprocessing Concept	x	x	x	x
Data Preprocessing Implementation & Execution	x	x	x	
Scone – Technical Basis of Logging	x			
WebChains Plugin – Code for actual Logging				x
Sequential Log File Visualization		x		
Categorization of Visited Pages		x		x
Analysis of Individual Daily Web Usage Patterns				x
Analysis of Technical Sessions				x
Manual Identification of 292 Intentional Sessions				x
Analysis of Revisits to Groups of Pages				x
SGUSs Concept and Implementation				x
Infrequent Revisits Analysis				x
Google Visits Analysis		x		
Multiple Window Analysis	x			

Table 9-2: Single contributions of the four authors to the web usage study. A cross indicates that this / these person(s) delivered major contributions to the specified task. This list is incomplete - tasks that are not mentioned in chapter 5 are not listed here.

9.6.2 SGUSs-Detection-Examples

The following eight examples illustrate the character of automatically detected *session groups with identical URI-subsets* (SGUSs). See page 195ff for definition and discussion. Times between sessions are rounded to the next full hour.

Example 1

The following details refer to example 1 discussed on page 198. One subject created the following 7 T-sessions that share 5 different URIs – the time between the first visits of the first and last contributing T-session was about 84 days.

The 5 shared URI that occurred in all sessions of this SGUS:

```
http://www.alldatasheet.com/
http://allicmall.com/
http://elenota.iele.polsl.gliwice.pl/en/search.php?szuk=tda&szukaj.x=0&szukaj.y=0&man=--all--&ile=50&start=0
http://www.datasheetarchive.com/752.html
http://katalog.elektroda.net/indx00-UA.html
```

Technical Sessions that belong to this SGUS (as explained on page 193f):

Session #20:	Duration:	20 min., 16 sec.
	Total number of visits:	65
	Recurrence rate in session:	52%
	Session begin:	Wed Nov 10 13:17:05 CET 2004
	Time to next SGUS session:	2 days, 3 hours.
Session #31:	Duration:	58 min., 15 sec.
	Total number of visits:	174
	Recurrence rate in session:	51%
	Session begin:	Fri Nov 12 16:58:17 CET 2004
	Time to next SGUS session:	2 days, 22 hours.
Session #44:	Duration:	18 min., 34 sec.
	Total number of visits:	73
	Recurrence rate in session:	51%
	Session begin:	Mon Nov 15 15:40:48 CET 2004
	Time to next SGUS session:	8 days, 2 hours.
Session #67:	Duration:	22 min., 40 sec.
	Total number of visits:	101
	Recurrence rate in session:	62%
	Session begin:	Tue Nov 23 17:54:43 CET 2004
	Time to next SGUS session:	9 days, 2 hours.
Session #95:	Duration:	27 min., 7 sec.
	Total number of visits:	98
	Recurrence rate in session:	49%
	Session begin:	Thu Dec 02 19:56:52 CET 2004
	Time to next SGUS session:	21 hours.

```

Session #99:  Duration:                20 min., 47 sec.
              Total number of visits:   96
              Recurrence rate in session: 55%
              Session begin:            Fri Dec 03 17:31:07 CET 2004
              Time to next SGUS session: 61 days, 2 hours.

Session #159: Duration:                31 min., 14 sec.
              Total number of visits:   95
              Recurrence rate in session: 49%
              Session begin:            Wed Feb 02 19:53:08 CET 2005

```

Example 2:

The following details refer to example 2 discussed on page 198. These two T-sessions contain 21 shared URIs – time between first visits of first and last T-session: almost 3 days.

The 23 different common URIs that occurred in both sessions of this SGUS:

```

http://www.alldatasheet.com/
http://www.eechain.com/datasheet/cndefault.asp
http://192.168.6.1/
http://www.eechain.com/datasheet/cnmfg.asp?page=2&keyword=S
http://allicmall.com/
http://www.google.com/webhp?sourceid=mozclient&ie=utf-8&oe=utf-8&q=
http://elenota.iele.polsl.gliwice.pl/en/search.php?szuk=tda&szukaj.x=0&szuk
aj.y=0&man=-all--&ile=50&start=0
http://www.datasheetarchive.com/752.html
http://fileshare.eshop.bg/?what=search
http://www.icbank.com/ENG/home/
http://membres.lycos.fr/etronics/datasheets/circuits/
http://www.radiolocman.com/comp/search-en.html
http://alldatasheet.com/datasheet-pdf/pdf/PHILIPS/TDA3508.html
http://fileshare.eshop.bg/?what=search2
http://smartdata.usbid.com/?userid=3004220
http://katalog.elektroda.net/indx00-UA.html
http://www.datasheetarchive.com/
http://katalog.elektroda.net/?d_start=00
http://www.digchip.com/datasheets/members_login.php
http://pdfs.cjb.net/
http://www.datadart.com/al/datasheets2.html

```

Technical Sessions that belong to this SGUS (as explained on page 193f):

```

Session #31:  Duration:                58 min., 15 sec.
              Total number of visits:   174
              Recurrence rate in session: 51%
              Session begin:            Fri Nov 12 16:58:17 CET 2004
              Time to next SGUS session: 2 days, 21 hours.

Session #44:  Duration:                18 min., 34 sec.
              Total number of visits:   73
              Recurrence rate in session: 51%
              Session begin:            Mon Nov 15 15:40:48 CET 2004

```

Example 3:

While working on the design of a client's website, one participant revisited 8 URIs in a second T-session after 5 days. He worked on the comparison of different exist-

ing site designs in order to create the customer's site. Thus, these revisits can be characterized as "content revisits" rather than "tool revisits". The 8 URIs are distributed over 7 different websites, which means that different bookmarks would have been necessary for a group revisitation, since no links connected the pages. The participant stated in the interview, that he did not know that he would come back later to these pages. Cumbersomely, he had to find the URIs again in different emails and handwritten notes.

2 T-sessions with 8 shared URIs - time between first visits of first and last T-session: 5 days, 15 hours, 40 min., 44 sec.

The 8 different common URIs that occurred in all sessions of this SGUS:

```
http://www.eveni.com/
http://www.acromediainc.com/
http://www.ebizsitedesigns.com/
http://kivs2001.informatik.uni-hamburg.de/
http://www.softconf.com/START/
http://www.conftool.net/
http://www.paperdyne.de/
http://vsys1.informatik.uni-hamburg.de/conftool.13/
```

Technical Sessions that belong to this SGUS (as explained on page 193f):

Session #293:	Duration:	36 min., 21 sec.
	Total number of visits:	23
	Recurrence rate in session:	39%
	Session begin:	Wed Jan 19 09:22:15 CET 2005
	Time to next SGUS session:	5 days, 15 hours.
Session #301:	Duration:	1 hours, 6 min., 53 sec.
	Total number of visits:	216
	Recurrence rate in session:	68%
	Session begin:	Tue Jan 25 01:02:59 CET 2005

Example 4:

Another subject was looking for an apartment during the study period. He revisited in a subsequent session after two and a half days 8 different URLs deriving from 5 different servers.

2 T-sessions with 8 shared URIs - time between first visits of first and last T-session: 2 days, 14 hours, 26 min., 14 sec.

The 8 different common URIs that occurred in both sessions of this SGUS:

```

http://www.google.de/
http://www.wg-gesucht.de/wg-zimmer-in-Hamburg-Alttona-Altstadt.139011.html
http://www.spiegel.de/
http://www.wg-gesucht.de/wg-zimmer-in-Hamburg.55.0.0.0.html
http://wohnboerse.asta.uni-hamburg.de/neu-
index.php?vslamp=0.8&db=wohnb&login=41dfcc2c&click=2
http://www.wg-gesucht.de/wg-zimmer-in-Hamburg-Eimsbuettel-
Eppendorf.135659.html
http://www.studenten-wg.de/mietangebot_235584.html
http://www.wg-gesucht.de/wg-zimmer-in-Hamburg.55.0.0.1.html

```

Technical Sessions that belong to this SGUS (as explained on page 193f):

Session #84:	Duration:	15 min., 26 sec.
	Total number of visits:	52
	Recurrence rate in session:	42%
	Session begin:	Mon Jan 10 22:31:16 CET 2005
	Time to next SGUS session:	2 days, 14 hours.
Session #86:	Duration:	4 hours, 29 min., 1 sec.
	Total number of visits:	516
	Recurrence rate in session:	43%
	Session begin:	Thu Jan 13 12:57:30 CET 2005

Example 5:

Some SGUSs were found in the logs of several participants that did not relate to one meaningful superior task, e.g. as a result of the habit to check news or to monitor the status of online auctions in breaks of the actual work. One participant, for example, shared three news and auction URIs in 8 different T-sessions over 50 days. Here, better support would be interesting even if the different visits do not belong to one enclosing subjective session.

8 T-sessions share 3 different URIs - time between first visits of first and last T-session: 49 days, 5 hours, 53 min., 34 sec.

The 3 different common URIs that occurred in all sessions of this SGUS:

```

http://www.spiegel.de/
http://pages.ebay.de/community/chat/
http://www.heise.de/newsticker/

```

Technical Sessions that belong to this SGUS (as explained on page 193f):

Session #39:	Duration:	32 min., 29 sec.
	Total number of visits:	60
	Recurrence rate in session:	33%
	Session begin:	Fri Aug 27 10:28:03 CEST 2004
	Time to next SGUS session:	3 days, 8 hours.
Session #58:	Duration:	26 min., 43 sec.

	Total number of visits:	81
	Recurrence rate in session:	40%
	Session begin:	Mon Aug 30 19:07:46 CEST 2004
	Time to next SGUS session:	51 min.
Session #59:	Duration:	21 min., 34 sec.
	Total number of visits:	64
	Recurrence rate in session:	33%
	Session begin:	Mon Aug 30 20:25:14 CEST 2004
	Time to next SGUS session:	14 days, 12 hours.
Session #143:	Duration:	1 hours, 26 min., 15 sec.
	Total number of visits:	88
	Recurrence rate in session:	32%
	Session begin:	Tue Sep 14 08:27:33 CEST 2004
	Time to next SGUS session:	3 hours.
Session #146:	Duration:	2 hours, 15 min., 18 sec.
	Total number of visits:	144
	Recurrence rate in session:	43%
	Session begin:	Tue Sep 14 12:38:05 CEST 2004
	Time to next SGUS session:	18 hours.
Session #149:	Duration:	1 hours, 14 min., 29 sec.
	Total number of visits:	94
	Recurrence rate in session:	41%
	Session begin:	Wed Sep 15 09:31:30 CEST 2004
	Time to next SGUS session:	22 hours.
Session #150:	Duration:	3 hours, 17 min., 58 sec.
	Total number of visits:	205 diffURLs: 105
	Session begin:	Thu Sep 16 08:47:50 CEST 2004
	Time to next SGUS session:	29 days, 4 hours.
Session #213:	Duration:	33 min., 54 sec.
	Total number of visits:	88
	Recurrence rate in session:	63%
	Session begin:	Fri Oct 15 16:21:37 CEST 2004

Example 6:

In this case, 2 T-sessions share 38 different URIs - time between first visits of first and last T-session: 4 days, 1 hour, 2 minutes.

The 38 different common URIs that occurred in all sessions of this SGUS:

```

http://www.n-tv.de/
http://www.baseb.de/?main=1&sub=2&det=1
http://www.autotask.com/solutions/for_internal_service_organizations.htm
http://www.baseb.de/?main=1
http://www.baseb.de/?main=2&sub=3
http://tour.celoxis.com/psa/servlet/ss?tpl=global/genericPage.wm&title=My+W
atched+Forums&tplname=my/mywatchedforums.wm
http://tour.celoxis.com/psa/servlet/user?bxn=umyhome&p_authenticate=true&p_
auth_login=mark&p_auth_password=celoxisl&p_client_tz=America/New_York
http://tour.celoxis.com/psa/servlet/user?bxn=usinoutstatus
http://tour.celoxis.com/psa/servlet/forum?bxn=mymy
http://www.centra.com/demovideo/personalDemo.asp
http://www.celoxis.com/html/pricing.php
http://www.vrtprj.de/
http://www.centra.com/demovideo/overview_flash.html
http://www.centra.com/demovideo/quicktours.asp

```

```

http://tour.celoxis.com/psa/servlet/folder?bxn=foview&p_fo_id=5
http://tour.celoxis.com/psa/servlet/forum?bxn=forumsfolder&p_fo_id=3
http://tour.celoxis.com/psa/servlet/user?bxn=umyhome&p_authenticate=true&p_auth_login=roberto&p_auth_password=celoxis1&p_client_tz=America/New_York
http://www.uni-hamburg.de/
http://www.welt.de/
http://www.carambalabs.com/ger/ed_team.html
http://www.mozilla.org/products/firefox/
http://www.businessengine.com/products/ben.html
http://www.vrtprj.de/products/products.html
http://www.vrtprj.de/products/
http://www.acos.com/
http://www.vrtprj.de/services/autotask.html
http://www.vrtprj.de/products/acosweb.html
http://www.businessengine.com/
http://www.vrtprj.de/products/businessenginetwork.html
http://www.vrtprj.de/products/baseb.html
http://www.businessengine.com/products/ben.asp
http://www.vrtprj.de/products/centraone.html
http://www.vrtprj.de/products/carambateamedition.html
http://www.vrtprj.de/products/celoxis.html
http://www.autotask.com/
http://www.kesys.de/
http://www.centra.com/demovideo/
http://www.centra.com/

```

Technical Sessions that belong to this SGUS (as explained on page 193f):

```

Session #43:  Duration:                3 hours, 38 min., 37 sec.
              Total number of visits:   293
              Recurrence rate in session: 47%
              Session begin:            Thu Oct 28 08:31:38 CEST 2004
              Time to next SGUS session: 3 days, 21 hours.

Session #44:  Duration:                5 hours, 54 min., 42 sec.
              Total number of visits:   497
              Recurrence rate in session: 49%
              Session begin:            Mon Nov 01 08:33:38 CET 2004

```

Example 7:

Here, 3 T-sessions share 8 different URIs - time between first visits of first and last T-session: 7 days, 8 hours, 16 min., 37 sec.

The 8 different common URIs:

```

http://www2.telegraaf.nl/
http://www.aegee.utwente.nl/~incognito/oud/agenda/deactiviteit.php?id=449
http://www.aegee.utwente.nl/~incognito/oud/webbers/reacties/addressa.php?soort=agenda&itemid=449
http://www.telegraaf.nl/common/templates/nav\_digidition\_frontpage\_flash.html
http://www.aegee.utwente.nl/~incognito/oud/navigate.php
http://www.aegee.utwente.nl/~incognito/oud/gentsnew.php
http://www.aegee.utwente.nl/~incognito/oud/agenda/agenda.php
http://www.nu.nl/

```

Technical Sessions that belong to this SGUS (as explained on page 193f):

Session #109: Duration: 3 hours, 11 min., 53 sec.
Total number of visits: 127
Recurrence rate in session: 24%
Session begin: Mon Feb 14 08:35:54 CET 2005
Time to next SGUS session: 1 days, 21 hours.

Session #119: Duration: 24 min., 21 sec.
16 Total number of visits: 37
Recurrence rate in session: 43%
Session begin: Wed Feb 16 08:41:36 CET 2005
Time to next SGUS session: 5 days, 8 hours.

Session #125: Duration: 23 min., 36 sec.
Total number of visits: 51
Recurrence rate in session: 25%
Session begin: Mon Feb 21 16:52:31 CET 2005

9.7 Formalia

9.7.1 Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die vorliegende Dissertation “Visualizing Web Sessions: Improving Web Browser History by a Better Understanding of Web Page Revisitation and a New Session- and Task-Based, Visual Web History Approach” eigenständig verfasst zu haben und keine anderen als die angegebenen Hilfsmittel verwendet zu haben.

Hamburg, 21. Juni 2007

Matthias Mayer