Peter King
Ethan V. Munson (Eds.)

LNCS 2023

Digital Documents: Systems and Principles

8th International Conference on Digital Documents and Electronic Publishing, DDEP 2000
5th International Workshop on the Principles of Digital Document Processing, PODDP 2000
Munich, Germany, September 2000, Revised Papers

Springer

Peter King Ethan V. Munson (Eds.)

Digital Documents: Systems and Principles

8th International Conference on Digital Documents and Electronic Publishing, DDEP 2000
5th International Workshop on the Principles of Digital Document Processing, PODDP 2000
Munich, Germany, September 13-15, 2000
Revised Papers

Springer

Volume Editors

Peter King
University of Manitoba, Department of Computer Science
Winnipeg, Manitoba R3T 2N2, Canada
E-mail: prking@cs.umanitoba.ca

Ethan V. Munson
University of Wisconsin - Milwaukee, Department of EECS
Milwaukee, WI 53201, USA
E-mail: munson@cs.uwm.edu

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): H.4.1, H.5.1, I.3.3, I.3.7, H.5.4, I.7, H.3

ISSN 0302-9743
ISBN 3-540-21070-9 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 10988053 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2023

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Preface

This volume contains the proceedings of two recent conferences in the field of electronic publishing and digital documents:

- DDEP 2000, the 8th International Conference on Digital Documents and Electronic Publishing, the successor conference to the EP conference series; and
- PODDP 2000, the 5th International Workshop on the Principles of Digital Document Processing.

Both conferences were held at the Technische Universität München, Munich, Germany in September 2000.

DDEP 2000 was the eighth in a biennial series of international conferences organized to promote the exchange of novel ideas concerning the computer production, manipulation and dissemination of documents. This conference series has attempted to reflect the evolving nature and usage of documents by treating digital documents and electronic publishing as a broad topic covering many aspects. These aspects have included document models, document representation and document dissemination, dynamic and hyper-documents, document analysis and management, and wide-ranging applications. The papers presented at DDEP 2000 and in this volume reflect this broad view, and cover such diverse topics as hypermedia structure and design, multimedia authoring techniques and systems, document structure inference, typography, document management and adaptation, document collections and Petri nets. All papers were refereed by an international program committee.

PODDP 2000 was designed to provide a forum for the discussion of the modeling of systems that process digital documents using theories and techniques from such fields as computer science, mathematics and psychology. The papers presented at PODDP 2000 appearing in this volume report on such diverse topics as abstract document structures and document data structures, techniques for document transformation, the applicability of UML (Unified Modeling Language) diagrams to document specifications, and automatic link generation. Again, all papers were refereed by an international program committee.

This volume also includes two papers that were previously accepted for the journal EPODD, Electronic Publishing, Origin Dissemination and Design. One of these papers contains a comparative evaluation of two common approaches to the electronic presentation of news, while the other paper describes an agent-based toolkit for finding and remembering information in a distributed environment of rapidly changing information sources. These two papers were reviewed by the editorial board of the journal.

The editors would like to thank the members of the PODDP and DDEP program committees for their considerable assistance in providing very thorough reviews of the submissions. We gratefully acknowledge the support of the conference sponsors: Software-Offensive Bayern, Comet Computer GmbH, the Institut

für Informatik Technische Universität München, and Springer-Verlag. Prof. Dr. Anne Brüggemann-Klein worked tirelessly as the chair of DDEP 2000 and also managed local arrangements for both meetings. We also express our sincere gratitude to Frau Evelyn Gemkow of the Technische Universität München, whose service as the conference secretary was much appreciated, and to Frau Diana Gross, Dr. Stefan Hermann and other members of the Rechnerbetriebsgruppe at TUM for the design of the Web pages and other on-site technical assistance, including Internet access.

December 2000

Peter King,
Ethan V. Munson

DDEP 2000 Organization

Steering Committee

Jacques André (INRIA/IRISA, Rennes, France)
David F. Brailsford (University of Nottingham, UK)
Heather Brown (University of Kent at Canterbury, UK)
Anne Brüggemann-Klein (Technische Universität München, Germany)
Richard Furuta (Texas A&M University, USA)
Rolf Ingold (University of Fribourg, Switzerland)
Peter King (University of Manitoba, Canada)
Robert A. Morris (University of Massachusetts, Boston, USA)
Marc Nanard (LIRMM, Montpellier, France)
Christine Vanoirbeek (Swiss Federal Institute of Technology, Lausanne, Switzerland)

Program Committee

Robert B. Allen (Bellcore, USA)
Jacques André (INRIA/IRISA, Rennes, France)
Charles Bigelow (Bigelow & Holmes, USA)
David F. Brailsford (University of Nottingham, UK)
Heather Brown (University of Kent, Canterbury, UK)
Anne Brüggemann-Klein (Technische Universität München, Germany)
Giovanni Coray (Swiss Federal Institute of Technology, Lausanne, Switzerland)
Anton Eliëns (Vrije Universiteit, Amsterdam, The Netherlands)
Hans-Peter Frei (UBILAB, Union Bank of Switzerland, Zurich, Switzerland)
Richard Furuta (Texas A&M University, USA)
Charles F. Goldfarb (Information Management Consulting, Saratoga, CA, USA)
Roger D. Hersch (Swiss Federal Institute of Technology, Lausanne, Switzerland)
Christoph Hüser (GMD IPSI, Darmstadt, Germany)
Rolf Ingold (University of Fribourg, Switzerland)
Pekka Kilpeläinen (University of Helsinki, Finland)
Peter King (University of Manitoba, Canada)
Michel Leonard (Université de Geneve, Switzerland)
Dario Lucarella (CRA-ENEL, Milan, Italy)
Pierre MacKay (University of Washington, USA)
Robert A. Morris (University of Massachusetts, Boston, USA)
Ethan V. Munson (University of Wisconsin-Milwaukee, USA)
Makoto Murata (Fuji Xerox Information Systems, Kawasaki, Japan)
Marc Nanard (LIRMM, Montpellier, France)
Erich Neuhold (GMD IPSI, and Technische Universität Darmstadt, Germany)
Richard Rubinstein (Human Factors International, USA)

Christine Vanoirbeek (Swiss Federal Institute of Technology, Lausanne, Switzerland)
Hans van Vliet (Vrije Universiteit, Amsterdam, The Netherlands)
Zhenkun Yang (Peking University, Beijing, China)
Annie Zaenen (Rank Xerox Research Centre, Grenoble, France)

PODDP 2000 Organization

Steering Committee

Derick Wood (Hong Kong University of Science and Technology), Chair
Anne Brueggemann-Klein (Technische Universität München, Germany)
Richard Furuta (Texas A&M University, USA)
Ethan V. Munson (University of Wisconsin-Milwaukee, USA)
Makoto Murata (Fuji Xerox Information Systems, Japan)
Charles Nicholas (University of Maryland, Baltimore County, USA)

Program Committee

Ethan V. Munson (University of Wisconsin-Milwaukee, USA)
Derick Wood (Hong Kong University of Science and Technology), Co-chair
Howard Blair (Syracuse University, USA)
Heather Brown (University of Kent, Canterbury, UK)
Anne Brueggemann-Klein (Technische Universität München, Germany)
Richard Furuta (Texas A&M University, USA)
Heikki Mannila (University of Helsinki, Finland)
Makoto Murata (Fuji Xerox Information Systems, Japan)
Charles Nicholas (University of Maryland, Baltimore County, USA)

Lecture Notes in Computer Science

For information about Vols. 1–2834

please contact your bookseller or Springer-Verlag

Vol. 2964: T. Okamoto (Eds.), Topics in Cryptology – CT-RSA 2004. Proceedings, 2004. XI, 387 pages. 2004.

Vol. 2957: P. Langendoerfer, M. Liu, I. Matta, V. Tsaoussidis (Eds.), Wired/Wireless Internet Communications. Proceedings, 2004. XI, 307 pages. 2004.

Vol. 2951: M. Naor (Eds.), Theory of Cryptography. Proceedings, 2004. XI, 523 pages. 2004.

Vol. 2949: R. De Nicola, G. Ferrari, G. Meredith (Eds.), Coordination Models and Languages. Proceedings, 2004. X, 323 pages. 2004.

Vol. 2946: R. Focardi, R. Gorrieri (Eds.), Foundations of Security Analysis and Design II. VII, 267 pages. 2004.

Vol. 2943: J. Chen, J. Reif (Eds.), DNA Computing. X, 225 pages. 2004.

Vol. 2930: F. Winkler, Automated Deduction in Geometry. VII, 231 pages. 2004. (Subseries LNAI).

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2004. (Subseries LNAI).

Vol. 2916: C. Palamidessi (Eds.), Logic Programming. Proceedings, 2003. XII, 520 pages. 2003.

Vol. 2914: P.K. Pandya, J. Radhakrishnan (Eds.), FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. Proceedings, 2003. XIII, 446 pages. 2003.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. Proceedings, 2003. XX, 512 pages. 2003. (Subseries LNAI).

Vol. 2911: T.M.T. Sembok, H.B. Zaman, H. Chen, S.R. Urs, S.H. Myaeng (Eds.), Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access. Proceedings, 2003. XX, 703 pages. 2003.

Vol. 2910: M.E. Orlowska, S. Weerawarana, M.M.P. Papazoglou, J. Yang (Eds.), Service-Oriented Computing - ICSOC 2003. Proceedings, 2003. XIV, 576 pages. 2003.

Vol. 2908: K. Chae, M. Yung (Eds.), Information Security Applications. XII, 506 pages. 2004.

Vol. 2906: T. Ibaraki, N. Katoh, H. Ono (Eds.), Algorithms and Computation. Proceedings, 2003. XVII, 748 pages. 2003.

Vol. 2905: A. Sanfeliu, J. Ruiz-Shulcloper (Eds.), Progress in Pattern Recognition, Speech and Image Analysis. XVII, 693 pages. 2003.

Vol. 2904: T. Johansson, S. Maitra (Eds.), Progress in Cryptology - INDOCRYPT 2003. Proceedings, 2003. XI, 431 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. Proceedings, 2003. XVI, 1075 pages. 2003. (Subseries LNAI).

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. Proceedings, 2003. XV, 504 pages. 2003. (Subseries LNAI).

Vol. 2901: F. Bry, N. Henze, J. Ma luszyński (Eds.), Principles and Practice of Semantic Web Reasoning. Proceedings, 2003. X, 209 pages. 2003.

Vol. 2900: M. Bidoit, P.D. Mosses (Eds.), Casl User Manual. XIII, 240 pages. 2004.

Vol. 2899: G. Ventre, R. Canonico (Eds.), Interactive Multimedia on Next Generation Networks. Proceedings, 2003. XIV, 420 pages. 2003.

Vol. 2898: K.G. Paterson (Eds.), Cryptography and Coding. Proceedings, 2003. IX, 385 pages. 2003.

Vol. 2897: O. Balet, G. Subsol, P. Torguet (Eds.), Virtual Storytelling. Proceedings, 2003. XI, 240 pages. 2003.

Vol. 2896: V.A. Saraswat (Eds.), Advances in Computing Science – ASIAN 2003. Proceedings, 2003. VIII, 305 pages. 2003.

Vol. 2895: A. Ohori (Eds.), Programming Languages and Systems. Proceedings, 2003. XIII, 427 pages. 2003.

Vol. 2894: C.S. Laih (Eds.), Advances in Cryptology - ASIACRYPT 2003. Proceedings, 2003. XIII, 543 pages. 2003.

Vol. 2893: J.-B. Stefani, I. Demeure, D. Hagimont (Eds.), Distributed Applications and Interoperable Systems. Proceedings, 2003. XIII, 311 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003. (Subseries LNAI).

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. Proceedings, 2003. X, 215 pages. 2003. (Subseries LNAI).

Vol. 2890: M. Broy, A.V. Zamulin (Eds.), Perspectives of System Informatics. XV, 572 pages. 2003.

Vol. 2889: R. Meersman, Z. Tari (Eds.), On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops. Proceedings, 2003. XIX, 1071 pages. 2003.

Vol. 2888: R. Meersman, Z. Tari, D.C. Schmidt (Eds.), On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Proceedings, 2003. XXI, 1546 pages. 2003.

Vol. 2887: T. Johansson (Eds.), Fast Software Encryption. IX, 397 pages. 2003.

Vol. 2886: I. Nyström, G. Sanniti di Baja, S. Svensson (Eds.), Discrete Geometry for Computer Imagery. Proceedings, 2003. XII, 556 pages. 2003.

Vol. 2885: J.S. Dong, J. Woodcock (Eds.), Formal Methods and Software Engineering. Proceedings, 2003. XI, 683 pages. 2003.

Vol. 2884: E. Najm, U. Nestmann, P. Stevens (Eds.), Formal Methods for Open Object-Based Distributed Systems. Proceedings, 2003. X, 293 pages. 2003.

Vol. 2883: J. Schaeffer, M. Müller, Y. Björnsson (Eds.), Computers and Games. XI, 431 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003. (Subseries LNAI).

Vol. 2881: E. Horlait, T. Magedanz, R.H. Glitho (Eds.), Mobile Agents for Telecommunication Applications. Proceedings, 2003. IX, 297 pages. 2003.

Vol. 2880: H.L. Bodlaender (Eds.), Graph-Theoretic Concepts in Computer Science. XI, 386 pages. 2003.

Vol. 2879: R.E. Ellis, T.M. Peters (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003. Proceedings, 2003. XXXIV, 1003 pages. 2003.

Vol. 2878: R.E. Ellis, T.M. Peters (Eds.), Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003. Proceedings, 2003. XXXIII, 819 pages. 2003.

Vol. 2877: T. Böhme, G. Heyer, H. Unger (Eds.), Innovative Internet Community Systems. VIII, 263 pages. 2003.

Vol. 2876: M. Schroeder, G. Wagner (Eds.), Rules and Rule Markup Languages for the Semantic Web. Proceedings, 2003. VII, 173 pages. 2003.

Vol. 2875: E. Aarts, R. Collier, E.v. Loenen, B.d. Ruyter (Eds.), Ambient Intelligence. Proceedings, 2003. XI, 432 pages. 2003.

Vol. 2874: C. Priami (Eds.), Global Computing. XIX, 255 pages. 2003.

Vol. 2871: N. Zhong, Z.W. Raś, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems. Proceedings, 2003. XV, 697 pages. 2003. (Subseries LNAI).

Vol. 2870: D. Fensel, K.P. Sycara, J. Mylopoulos (Eds.), The Semantic Web - ISWC 2003. Proceedings, 2003. XV, 931 pages. 2003.

Vol. 2869: A. Yazici, C. Şener (Eds.), Computer and Information Sciences - ISCIS 2003. Proceedings, 2003. XIX, 1110 pages. 2003.

Vol. 2868: P. Perner, R. Brause, H.-G. Holzhütter (Eds.), Medical Data Analysis. Proceedings, 2003. VIII, 127 pages. 2003.

Vol. 2866: J. Akiyama, M. Kano (Eds.), Discrete and Computational Geometry. VIII, 285 pages. 2003.

Vol. 2865: S. Pierre, M. Barbeau, E. Kranakis (Eds.), Ad-Hoc, Mobile, and Wireless Networks. Proceedings, 2003. X, 293 pages. 2003.

Vol. 2864: A.K. Dey, A. Schmidt, J.F. McCarthy (Eds.), UbiComp 2003: Ubiquitous Computing. Proceedings, 2003. XVII, 368 pages. 2003.

Vol. 2863: P. Stevens, J. Whittle, G. Booch (Eds.), "UML" 2003 - The Unified Modeling Language. Proceedings, 2003. XIV, 415 pages. 2003.

Vol. 2860: D. Geist, E. Tronci (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 2003. XII, 426 pages. 2003.

Vol. 2859: B. Apolloni, M. Marinaro, R. Tagliaferri (Eds.), Neural Nets. X, 376 pages. 2003.

Vol. 2857: M.A. Nascimento, E.S. de Moura, A.L. Oliveira (Eds.), String Processing and Information Retrieval. Proceedings, 2003. XI, 379 pages. 2003.

Vol. 2856: M. Smirnov (Eds.), Quality of Future Internet Services. IX, 293 pages. 2003.

Vol. 2855: R. Alur, I. Lee (Eds.), Embedded Software. Proceedings, 2003. X, 373 pages. 2003.

Vol. 2854: J. Hoffmann, Utilizing Problem Structure in Planing. XIII, 251 pages. 2003. (Subseries LNAI).

Vol. 2853: M. Jeckle, L.-J. Zhang (Eds.), Web Services - ICWS-Europe 2003. VIII, 227 pages. 2003.

Vol. 2852: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 509 pages. 2003.

Vol. 2851: C. Boyd, W. Mao (Eds.), Information Security. Proceedings, 2003. XI, 453 pages. 2003.

Vol. 2849: N. García, L. Salgado, J.M. Martínez (Eds.), Visual Content Processing and Representation. Proceedings, 2003. XII, 352 pages. 2003.

Vol. 2848: F.E. Fich (Eds.), Distributed Computing. Proceedings, 2003. X, 367 pages. 2003.

Vol. 2847: R.d. Lemos, T.S. Weber, J.B. Camargo Jr. (Eds.), Dependable Computing. Proceedings, 2003. XIV, 371 pages. 2003.

Vol. 2846: J. Zhou, M. Yung, Y. Han (Eds.), Applied Cryptography and Network Security. Proceedings, 2003. XI, 436 pages. 2003.

Vol. 2845: B. Christianson, B. Crispo, J.A. Malcolm, M. Roe (Eds.), Security Protocols. VIII, 243 pages. 2004.

Vol. 2844: J.A. Jorge, N. Jardim Nunes, J. Falcão e Cunha (Eds.), Interactive Systems. Design, Specification, and Verification. XIII, 429 pages. 2003.

Vol. 2843: G. Grieser, Y. Tanaka, A. Yamamoto (Eds.), Discovery Science. Proceedings, 2003. XII, 504 pages. 2003. (Subseries LNAI).

Vol. 2842: R. Gavaldá, K.P. Jantke, E. Takimoto (Eds.), Algorithmic Learning Theory. Proceedings, 2003. XI, 313 pages. 2003. (Subseries LNAI).

Vol. 2841: C. Blundo, C. Laneve (Eds.), Theoretical Computer Science. Proceedings, 2003. XI, 397 pages. 2003.

Vol. 2840: J. Dongarra, D. Laforenza, S. Orlando (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. Proceedings, 2003. XVIII, 693 pages. 2003.

Vol. 2839: A. Marshall, N. Agoulmine (Eds.), Management of Multimedia Networks and Services. Proceedings, 2003. XIV, 532 pages. 2003.

Vol. 2838: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Knowledge Discovery in Databases: PKDD 2003. Proceedings, 2003. XVI, 508 pages. 2003. (Subseries LNAI).

Vol. 2837: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Machine Learning: ECML 2003. Proceedings, 2003. XVI, 504 pages. 2003. (Subseries LNAI).

Vol. 2836: S. Qing, D. Gollmann, J. Zhou (Eds.), Information and Communications Security. Proceedings, 2003. XI, 416 pages. 2003.

Vol. 2835: T. Horváth, A. Yamamoto (Eds.), Inductive Logic Programming. Proceedings, 2003. X, 401 pages. 2003. (Subseries LNAI).

Table of Contents

DDEP: Links and Structure

DDEP: Typrography and Authoring

DDEP: Management and Adaption

PODDP

A Link-Oriented Comparison of Hyperdocuments and Programs

Heather Brown[1], Peter Brown[1], Les Carr[2], Wendy Hall[2], Wendy Milne[1], and Luc Moreau[2]

[1]Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK
{H.Brown,P.J.Brown,W.Milne}@exeter.ac.uk

[2]Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK
{L.A.Carr,W.Hall,L.Moreau}@ecs.soton.ac.uk

Abstract. There are parallels between the construction of programs and the construction of hypertexts, and in particular between the abstractions available to the application programmer and those available to the hypertext author. In this paper we look at the distinctive element of the hypertext medium, the link, and discuss its possible programming language analogs. We go on to examine programming language abstractions that could be usefully employed by hypertext authors to control the complexity of the systems which they are engaged in building.

1 Introduction, Background, and Assumptions

Following Dijkstra's famous article 'Goto considered harmful' [8], written in 1968, the goto statement has been deprecated in most programming languages. Remaining at the lowest abstraction levels, such as assembly languages, or used for efficiency reasons, it is masked by higher-level abstractions (e.g. selections, procedure calls, method invocations and continuations). On the other hand, the hypertext link, which has been characterized as a goto [7], has remained in use in hypertext. Indeed many people see it as the essence of hypertext; most of the definitions of hypertext originally given by Nielsen [16] centre around linking.

Although programming involves many variations on the goto which are safer for programming-in-the-small (if/then, case, iterations) and -in-the-large (procedures, modules, class libraries), the simple link remains the principal tool for the hypertext engineer. The complexity inherent in the unconstrained use of this primitive construct is one of the main issues in hypertext design.

This paper tries to analyse the apparent clash of practice. It also covers a wider issue: many authors have extended the goto/link analogy by likening the authorship of a hyperdocument to the task of writing programs, or, at a higher level, have mapped out a discipline of hypermedia engineering to match software engineering [13]. These comparisons can be valuable because hyperdocument authoring is a young discipline

P. King and E.V. Munson (Eds.): DDEP-PODDP 2000, LNCS 2023, pp. 1–12, 2004.

compared with the discipline of producing programs; if, by drawing parallels with programming, we can gain new insights into hyperdocument authoring, there are big potential gains. We must ensure, however, that the parallels are valid ones, and this paper tries to help. Our main focus, reflected in the title, is at the comparatively low level of links, rather than the higher levels of structuring and engineering. Maintaining this focus, we look at ways of modelling links that are more expressive than a simple goto and look at the difference between the static and dynamic aspects of hypertext construction. Finally we list a number of control abstractions that have been used by software engineers and consider how they may help the hypertext author.

1.1 Assumptions

In order to make this paper simpler, we shall fix some of the objects we are talking about. We shall assume that the hypertext is represented in HTML and viewed on a web browser. An HTML document may host scripting components, applets and various kinds of dynamic event handlers, but if we discuss hyperdocuments that involve bits of program this will inevitably muddy our discussion. Thus we shall confine our discussion to static hypertext: no pieces of Java, no CGI scripts, no cookies, etc. Our HTML hyperdocument will, of course, consist of a number of pages, and these will in general link to pages outside the current hyperdocument.

Notwithstanding our use of HTML as a basis for example, we will refer to hypertext systems other than the World Wide Web, since many of these are more developed in the abstractions they provide. In particular HTML essentially only offers one type of link, though there is a potentially extremely rich set of link types that hypertext systems may offer [4,14].

1.2 The Author and the End-User

If we start on the programming side, the two important parties are the author(s) and the end-user(s). The author creates the program, which may be a module in a much larger program, and the end-user executes the program. The author's world is a long way away from the end-user's. Indeed the end-user is normally unaware of the nature of the source code, and whether it contains any gotos. The goto concept, and indeed all concepts of program structuring, just apply to the author's source code world.

In the hypertext world, the author prepares a document. The end-user reads the document the author has built. However the end-user's world is much closer to the author's than is the case for a program. In particular the links provided by the author are directly visible to the end-user. Thus, reflecting these two levels, we can draw two comparisons:

(a) between gotos in programs and the complete set of all links specified by hypertext authors, or

(b) between gotos in programs and the actual set of links used during a browser session by hypertext end-users.

We believe that (a) is the closer comparison, but (b) still deserves attention since the goto is principally happening to the end user. The text does not 'go' anywhere, instead there is an intuitive understanding that the user has 'travelled', hence the common reference to 'navigating' or 'surfing' the Web. The role of the author is to specify the complete set of gotos, i.e. to determine the possible navigational choices from which the user chooses the actual set.

2 Alternatives to the Goto Model

Our first point is that if one wishes to liken a hypertext link to a programming language concept, there are several alternatives to the goto model. We will discuss three of them here: we will call them the link-is-data-reference model, the link-is-procedure-call model and the link-is-a-continuation model.

The first model, the link-is-data-reference model, is simple. In this, a hyperdocument is likened to the data part of a program, not the executable part. Each hyperdocument page is likened to a particular data structure (or to an object in OO technology), and links are just references to other data structures. As an example, if the hypertext page just consists of some text T1, followed by a link L, followed by further text T2, then, using Java notion together with a very simple document object model, this is likened to the programming language data structure:

```
final String T1 = "If you are interested, please";
final Link L = new Link("click here",
       "http://site.org/data.html");
final String T2 = "for more information";
final Page P = new Page(T1, L, T2);
```

This is a simplistic view of a Web page, the Web Consortium's DOM standard [19] is a more complete mechanism for treating a hypertext page as just such a simple data structure. The effect of this from our point of view is to reduce a link to an undistinguished component of the data structure, and requires the navigation behaviour (and the rendering activity) to be specified by external semantics in the form of stylesheet data or scripted functions.

Our second model, the link-is-procedure-call model, is closer to the goto model in that it relates to the executable part of a program. Although a Web server is stateless and is therefore unaffected by the user's choice of link navigation (without the use of cookies or explicitly programmed work-arounds), each browser maintains both a linear history and a stack of previously-visted pages together with a Back button. Given a Back facility, a link is arguably a specification of a procedure call, not a goto at all. A model, reflecting the analogy that each hyperdocument page is a procedure that potentially calls other procedures, is that a page can be likened to a procedure with the following body: