

Versioning hypermedia

Fabio Vitali

University of Bologna

Department of Computer Science Web: <http://www.cs.unibo.it/>

Mura A. Zamboni, 7

I-40127 Bologna BO Italy

Email: fabio@cs.unibo.it

Web: <http://www.cs.unibo.it/~fabio/>

Abstract: Keeping multiple versions of the same electronic artifact is a necessity in many authoring fields, and a serious advantage in all of them. Hypermedia adds to that the issue of relationship management. This poses a few additional problems, especially conceptual ones, but it also provides a reliable and safe solution for the well-known problem of the referential integrity of links. The field of hypermedia has dealt with versioning issues for a long time, since Xanadu considered it a fundamental mechanism for its inner workings. Newer systems, and an important protocol for the WWW, WebDAV, constitute modern approaches to the problem.

Introduction

Versioning is the management of multiple copies of the same evolving resource, captured at different stages of its evolution. Versioning is a well-known technique whenever there is data to be authored: it provides for the verification of progress in authoring, it guarantees fail-safe baselines for exploratory changes, it supports verification and comparison of the individual efforts in a multi-authoring situation.

Yet, versioning adds (sometimes heavily) to the authoring effort, in terms both of system resources, and most importantly of the conceptual overhead for authors: dealing with versions affects the straightforwardness of the work with quite a number of chores unrelated to the main writing task. Just like backing up data, versioning requires constant awareness for just possible usefulness, and its advantages are sometimes appreciated only when it is already too late.

It is not surprising therefore that versioning is often seen as an optional commodity, and sometimes as a burden, rather than as an essential apparatus of the authoring environment. Only in some specialized and highly sophisticated authoring communities, such as software engineering and database management, has versioning become part of the routine chores of the practitioners.

In hypermedia, insofar as it is viewed as an authoring situation, the problems connected to version management have been frequently examined and discussed. Of course, most of the problems that apply to other authoring environments also apply to hypermedia. Yet, there are peculiarities in versioning inter-linked content, that at the same time pose some new problems, and propose an elegant solution to an important issue, referential integrity of links, that is specific of hypermedia. This may in part be the reason traditional approaches to versioning have been customized to the requirements specifics of hypermedia.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or permissions@acm.org

Several important systems throughout the long history of hypermedia have discussed, implemented, or even relied on versioning functionalities, from Nelson's Xanadu [Nelson 1987] to the current IETF standardization effort WebDAV [Slein 1998], ranging through well-known systems such as SEPIA ([Haake 1992] and [Haake 1994a]), Hyperform [Wiil 1992b], and Microcosm [Melly 1995].

This short presentation is thus meant to introduce the basic issues of versioning hypermedia, and a brief and necessarily incomplete roundup of the systems and proposals that have made the history of this research field.

Advantages of versioning for hypermedia

In many cases, being able to access older versions of a resource (e.g., a set of related documents) is downright necessary, for instance in the software development process, where it is important to maintain (i.e. verify and modify) the deployed releases even while someone else is creating a new one, or in most legal systems, where the law to be applied is the one that was valid at the moment of the affected event, although its content may have changed in the meantime.

In situations where the authoring is managed and controlled within a workflow, the accounting and verification of the authoring activities provide additional motivations for accessing older versions of the artifact: in these cases, it is useful of comparing the current state of the resource with older versions, in order to determine the changes, keep a record of them, and evaluate the progress of the development.

Furthermore, versioning helps in exploratory authoring: keeping a good and reliable baseline version makes authors more confident in doing experiments and trying out new development paths with their documents, even if these paths eventually turn out to be unviable or unacceptable, since they can return to the baseline at any time and re-start experimenting from there.

And, obviously, distributed and collaborative authoring may profit from version support: verification and evaluation of others' contribution is eased by the possibility to compare their work with previous baseline versions of the resource; furthermore, if the chain of subsequent versions is allowed to fork into independent branches, then multiple authors can work on the same resource at the same time with no risk of overwriting, since each author would work on an independent version of the same resource.

It has been noted (for instance by Wiil and Leggett [Wiil 1993a]) that in many creative authoring situations (including hypermedia) long transactions are crucial, because operations in authoring environments are often long-lived. Relying on locking for concurrency control, even fine-grained user-controlled locking, may be excessively exacting in certain situations, and it may be preferable to allow the development path of a resource to diverge into two or more independent branches, that can be merged into a single, final state in a later stage of the development.

Branching versions could even allow collaboration to emerge without planning: emergent collaboration [Maioli 1994] happens when readers of a published resource provide the original authors with suggestions, modifications, additions, additional links, etc. which, when put into a separate branch of the official version tree of the resource, can be accepted and accessed by readers without requiring integration or officialization by the main authors.

Versioning provides an additional advantage for hypermedia, providing an easy solution to an old problem of hypermedia: that of the referential integrity of links. In the words of Ted Nelson ([Nelson1987], but see also [Nelson 1999]):

But if you are going to have links you really need historical backtrack and alternative versions. Why? Because if you make some links to another author's document on Monday and its author goes on making changes, perhaps on Wednesday you'd like to follow those links into the present version. They'd better still be attached to the right parts, even though the parts may have moved.

The problem has been recently discussed by Davis ([Davis 1998] and [Davis 1999]), and can be summarized as follows:

- In many situations, it is useful to link not just whole resources, but specific locations within the resource (the so called *point-to-point links*).
- There are very good reasons to store links outside of the resources they connect, rather than embedded inside them *à la HTML*. For instance, external links allow linking into resources that we have no write access to, that are stored in a read-only medium, or that have been heavily linked by other users for other purposes we are not interested in. Furthermore, external links allow private links, specialized trails, and multiple independent link sets to be created for the same document set. Hypertext systems that implement external links are too numerous to mention, but it is worth noting that XLink ([DeRose 1999a] and [DeRose 1999]), the linking protocol for the XML language, also allows external links.
- Whenever a linked resource changes, external links risk pointing to the wrong place because of the changes themselves: whatever method is used for referring to a specific location of a resource (e.g., counting, searching or naming), there may be a change that messes up the reference.
- Fixing dangling references can be performed either by having a human retrieve the new position and update the reference, or by applying a "best bet" heuristics, whereby the current link end-point is determined by finding the most similar content to the old end-point, or by position tracking, that is, by following change after change the evolution of the references, thus determining their current positions.

Of these solutions, only position tracking requires no human effort and guarantees a correct solution in all cases. Position tracking, on the other hand, requires that the resources being linked are versioned, since it implies accessing subsequent versions of the same resource, comparing them in a sufficiently fine-grained detail, and following version after version the evolution of the relevant positions.

Versioning issues in hypermedia

The adoption of a versioning mechanism within a hypertext system raises several important issues.

An important concept is that of version models. Haake and Hicks [Haake 1996] identified two basic version models: *state-based versioning* maintain the version of an individual resource, while *task-based versioning* focus on tracking versions of complex systems as a whole. These concepts are similar to those of state-based and change-based versioning as known in software engineering [Conradi 1998]. State-based versioning does not support the tracking of a set of changes involving several components of a hypertext network, while task-based approaches provide system support for maintaining the relationships between versions of resources that have been changed in a coordinated manner during the performance of a task. Holistic, task-based approaches to versioning are especially sensible in hypermedia, given the complex, multi-dimensional aspects of modifying a complex hypermedia network.

Østerbye [Østerbye 1992] raises a pair of important structural problems in versioning hypermedia:

- the immutability of frozen versions
- the versioning of links

Intrinsic to any version model is the fact that older versions of a resource are *frozen*, that is, they cannot be changed without creating a new version of the resource. Yet it may be useful to allow frozen versions to have new links (for instance, annotations or comments) coming from and going to them without necessarily creating a new version of the resource. At the same time, some links are substantially part of the resource itself, and thus their modification should definitely require the creation of a new version of the whole resource. Depending on their meaning and their

role, therefore, the creation of some new links may or may not imply the creation of a new version of the resource. External link sets offer a solution to this problem: when creating a new version of a resource, the author would also specify the set of substantial links (the ones that if modified would create a new version of the resource), while all other links would be considered as annotation links (and would not require a new version of the resource if changed or incremented).

If links are external to the resources they connect, the issue of link versioning must be considered: given a link from A to B, what happens to it when, for instance, B is changed under the control of a versioning system? Does it follow through to the current version of the resource, and no link points to the old version of B any longer? Does it generate a new link pointing to the new version of the resource, so that we end up with two links from A, going to both the old and the new version of B? Or does it create a two versions of the whole hypermedia network, one in which the link goes from A to B (old version), and another in which the *same* link goes from A to B (new version)?

Differently from other fields, in hypermedia all these solutions could be valid and preferable over each other depending on the task being performed. Thus, the final decision is better left to the authors in a case-by-case fashion. This leads to noteworthy cognitive problems such as version freezing and element selection (also discussed in [Østerbye 1992]). The issue of version freezing concerns the complexity of deciding the set of resources to be frozen in a stable state after an editing session, and of actually performing the freeze. The issue of element selection arises whenever there is a link to a resource of which there are more than one version, and there is no pre-defined policy as to which version of the resource should the link point to. Here, both sending the reader to a default version, and asking the reader to choose the version seem cumbersome and awkward solutions.

A brief history of versioning in hypermedia

Versioning has been a topic of hypermedia since a very early age. Version management is intrinsic and fundamental to the inner workings of the Xanadu system [Nelson 1987]. Xanadu proposes a peculiar way to organize the data, called the *Xanalogical storage* [Nelson 1999], where the documents (the minimal structure of the system) either actually *contain* their content (*native bytes*), or *refer to* it by inclusion from other documents (*included bytes*). In Xanadu, versioning is at the same time an immediate functionality of the system (a new version of a document is a new document that includes all the parts of a document that were present also in the previous version, and that has as native bytes all the new data) and a requirement: in Xanadu inclusions refer to their end-points by offsets, so that any change to the content of a document would corrupt the very structure of the inclusions unless exact tracking of the documents' evolution is activated through versioning. Later in time, both RHYTHM [Maioli 1993] and Palimpsest [Durand 1993] proposed solutions similar to the Xanalogical storage, heavily relying on versioning for the management of correct inclusions.

After the experience of the PIE system [Goldstein 1984] and Halasz's powerful keynote address to the Hypertext '87 Conference [Halasz 1988], where versioning was mentioned as one of the main open issues in the hypermedia field, many researchers set out to study the subject: CoVer ([Haake 1992] and [Haake 1994a]) is a contextual version server that can provide both state-based and task-based version models for the SEPIA hypermedia authoring system; within HB3 [Hicks 1993] and Hyperform [Wiil 1992b] researchers concentrated instead on abstracting the concept of version management from the actual hypertext systems they were going to provide such service for. HyperClas [Dattolo 1996] proposed a fully distributed and cooperative approach to versioning based on autonomous agents.

In 1994 and 1995 two versioning workshops ([Durand 1994] and [Hicks 1995]) helped to further shape the field, examining aspects such as link selection, conceptual overhead in version freezing, and support for collaboration in distributed hypermedia. Among the hypertext systems that discussed the implementation of some kind of version support we will also include Microcosm [Melly 1995] and Chimera [Whitehead 1994].

The advent of the World Wide Web introduced the new challenge of adding versioning functionalities to it. Vitali and Durand [Vitali 1995] proposed VTML (Versioned Text Markup language), a markup language to express change operations for WWW documents, in particular HTML. HTML 4.0 [Raggett 1999] includes two new tags,

INS and DEL, that are meant to express changes from previous versions of the same document (e.g., in legal texts); unfortunately these tags, being part of the markup language, cannot express changes in the markup itself (e.g., that two paragraphs have been joined, that a link destination has been changed, etc.) or changes that disrupt the correct nesting of the markup.

A newer activity is that of the WebDAV specifications. WebDAV is an IETF working group devoted to extend the HTTP protocol to support distributed authoring. Since it was felt that versioning would play an important role in the management of distributed collaboration over documents available from an HTTP server, versioning was made part of the requirements of the WebDAV group [Slein 1998] and will be covered by a forthcoming specification.

Conclusions

Hypermedia authoring shares many problems with other creative authoring environments. The issues arising from complex project management, support for multi-authored documents, and collaboration in hypermedia are similar to the corresponding ones in many other fields. Hypermedia differs from other fields because of the management of explicit relationships among the resources being managed.

Versioning hypermedia presents a few new problems because of the management of ad hoc relationships among versioned resources. On the other hand, besides many more advantages, versioning provides an easy and safe solution to the well-known problem of referential integrity of links.

It may seem overkill to advocate versioning just to make sure that point-to-point links do not refer a few bytes off the correct place after a few changes. Yet the real cases where the reliable tracking of positions are of paramount importance are still ahead of us: we haven't understood yet how a Xanalogical storage (i.e., how the virtualization of content through the systematic use of inclusion references of chunks of other possibly virtual documents [Nelson 1999]) may change our view of the creative process. Yet, for Xanalogical storage to be really working, it is necessary that a forced, automatic, transparent mechanism of versioning of data is put in place. No approximate, good-enough solutions for the management of changing references can conceivably be considered acceptable in this case. A sufficiently fine-grained versioning model for the electronic resources need to be implemented and used systematically.

References

- [Brailsford 1999] David F. Brailsford. "Separable hyperstructure and delayed link binding" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.
- [Conradi 1998] Reidar Conradi and Bernhard Westfechtel. "Version Models in Software Configuration Management" in ACM Computing Surveys, 30(2), 232-282, June 1998.
- [Dattolo 1996] Antonina Dattolo and Vincenzo Loia. "Collaborative Version Control in an Agent-based Hypertext Environment" in Information Systems, 21(2), 127-145, 1996.
- [Davis 1998] Hugh C. Davis. "Referential Integrity of Links in Open Hypermedia Systems" in Proceedings of ACM Hypertext '98, Pittsburgh, PA, 207-216, June 1998.
- [Davis 1999] Hugh C. Davis. "Hypertext Link Integrity" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.
- [DeRose 1999a] Steven J. DeRose, David Orchard, and Ben Trafford. "XML Linking Language (XLink)" (Working Draft 26 July 1999). Cambridge, Massachusetts: World Wide Web Consortium. [Online: <http://www.w3.org/1999/07/WD-xlink-19990726>; <http://www.w3.org/TR/WD-xml-link>], 1999.

[DeRose 1999] Steven J. DeRose. "XML Linking" in ACM Computing Surveys, Symposium on Hypertext and Hypermedia, 1999.

[Durand 1993] David G. Durand. "Cooperative Editing without Synchronization" in ACM Hypertext '93 Workshop on Hyperbase Systems (Seattle, WA), Technical Report n. TAMU-HRL 93-009, Hypertext Research Lab, Texas A&M University, College Station TX, 1993.

[Durand 1994] David G. Durand, Anya Haake, David Hicks, and Fabio Vitali (editors). Proceedings of the ECHT '94 Workshop on Versioning in Hypertext Systems, Technical Report of the Computer Science Department, Boston University, 95-01. [Online: <http://www.cs.bu.edu/techreports/95-001/Home.html>] and Arbeitspapiere der GMD 894, GMD-IPSI, Darmstadt, Germany, September 1994.

[Goldstein 1984] Ira Goldstein and Daniel Bobrow. "A Layered Approach to Software Design" in Interactive Programming Environments, D. Barstow, H. Shrobe, E. Sandewell (editors), McGraw Hill, 387-413, 1984.

[Haake 1992] Anya Haake. "CoVer: a Contextual Version Server for Hypertext Applications" in Proceedings of the ACM Conference on Hypertext (ECHT '92), Milano, Italy, 43-52, December 1992.

[Haake 1994a] Anya Haake. "Under CoVer: the Implementation of a Contextual Version Server for Hypertext Applications" in Proceedings of the ACM European Conference on Hypermedia Technology (ECHT '94), Edinburgh, Scotland, 81-93, September 1994.

[Haake 1996] Anya Haake and David Hicks. "VerSE: Towards Hypertext Versioning Styles" in Proceedings of ACM Hypertext '96, Washington DC, 224-234, March 1996.

[Halasz 1988] Frank G. Halasz. "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems" in Communications of the ACM (CACM), 31(7), 836-852, July 1988.

[Hicks 1993] David Hicks. A Version Control Architecture for Advanced Hypermedia Environments. PhD Dissertation, Department of Computer Science, Texas A&M University, College Station TX, 1993.

[Hicks 1995] David Hicks, Anya Haake, David G. Durand, and Fabio Vitali (editors). Proceedings of the ECSCW '95 Workshop on the Role of Version Control in CSCW Applications, Technical Report of the Computer Science Department, Boston University, 96-06, [Online: http://www.cs.bu.edu/techreports/96-009-ecscw95-proceedings/Book/proceedings_txt.html], 1995.

[Maioli 1993] Cesare Maioli, Stefano Sola, and Fabio Vitali. "Wide-Area Distribution Issues In Hypertext Systems" in Proceedings of ACM SIGDOC '93, Kitchener, Canada, 185-197, 1993.

[Maioli 1994] Cesare Maioli, Stefano Sola, and Fabio Vitali. "The Support for Emergence of Collaboration in a Hypertext Document System" in ACM CSCW'94 Workshop on Collaborative Hypermedia Systems, Chapel Hill, NC, GMD Studien n. 239, 1994.

[Melly 1995] Mylene Melly and Wendy Hall. "Version Control in Microcosm" in ECSCW '95 Workshop on the Role of Version Control in CSCW Applications, Stockholm, Sweden, 1995.

[Nelson 1987] Theodor Helm Nelson. Literary Machines, Edition 87.1, Sausalito Press, 1987.

[Nelson 1999] Theodor Helm Nelson. "Xanalogical Media: Needed Now More Than Ever" in ACM Computing Surveys Symposium on Hypertext and Hypermedia, 1999.

[Østerbye 1992] Kasper Østerbye. "Structural and Cognitive Problems in Providing Version Control for Hypertext" in Proceedings of the ACM Conference on Hypertext (ECHT '92), Milano, Italy, 33-42, December 1992.

[Raggett 1999] Dave Raggett, Arnaud Le Hors and Ian Jacobs, (editors). HTML 4.0 Specification, W3C Recommendation, [Online: <http://www.w3.org/TR/REC-html40/>], December 1999.

[Slein 1998] Judith Slein, Fabio Vitali, E. James Whitehead, Jr., and David G. Durand. "Requirements for Distributed Authoring and Versioning on the World Wide Web" in ACM StandardView 5(1), 17-24; Also published as RFC 2291, February 1998, IETF, [Online: <http://www.ics.uci.edu/pub/ietf/webdav/requirements/rfc2291.txt>], 1998.

[Vitali 1995] Fabio Vitali and David G. Durand. "Using Versioning to Provide Collaboration on the WWW" in The World Wide Web Journal 1(1), O'Reilly, 37-50, 1995.

[Whitehead 1994] E. James Whitehead. "A Proposal for Versioning Support for the Chimera System" in ECHT '94 Workshop on Versioning in Hypertext Systems, Edinburgh, Scotland, September 1994.

[Wiil 1992b] Uffe K. Wiil and John J. Leggett. "Hyperform: Using Extensibility to Develop Dynamic, Open, and Distributed Hypertext Systems" in Proceedings of the ACM Conference on Hypertext (ECHT '92), Milano, Italy, 251-261, December 1994.

[Wiil 1993a] Uffe K. Wiil and John J. Leggett. "Concurrency Control in Collaborative Hypertext Systems" in Proceedings of ACM Hypertext '93, Seattle, WA, 14-24, November 1993.