Comparing Spatial Hypertext Collections

J. Nathan Matias
World University Project
9a Bridge Street
Cambridge, UK CB2 1UB
j.n.matias@cantab.net

David P. Williams
University of Cambridge
Jesus College
Cambridge, UK CB5 8BL
dw341@cam.ac.uk

ABSTRACT

This paper proposes an approach to comparison of spatial hypertext collections which avoids becoming entangled in complexities of version management and merging. We also propose and illustrate principles for presenting comparisons of spatial hypertext without losing important implicit information.

We argue that multiple view options, distinct areas for different collections, and dependency lists are all necessary if comparison is to retain the kinds of meaning fundamentally important to spatial hypertext.

Categories and Subject Descriptors

H5.4 Hypertext/Hypermedia User Issues
H5.3 Group and Organization Interfaces Theory and Models

General Terms: Design, Human Factors, Theory.

Keywords: Spatial hypertext, comparison, collaboration, model merging, version management, Tinderbox, ShyWiki, siDiff, VUE, VKB, Webspiration

1. INTRODUCTION

The problem of hypertext comparison, an immediately-obvious component of any multi-user or version-managed system, is nonetheless still poorly understood. This is especially the case for spatial hypertext. To address this, we provide a simple approach to calculating and presenting spatial hypertext comparisons, explaining a key component to larger problems within spatial hypertext version management.

Since spatial hypertext implies meaning in the visual arrangement of information,[1] a versioning system for spatial hypertext must account for these visual features. Meaningful implications could be encoded in shapes, colours, borders, lines, arrows, regions of visual space, proximity of visual elements, and labels affixed to any of the above. Furthermore, spatial hypertext systems sometimes present information using multiple views. Each individual view could present important information which may not be presented by other views.

Managing versions of spatial hypertext therefore requires a special attention to visual information in ways not discussed in prior scholarship. Previous approaches have modeled the histories of objects, text, transclusions, and links, but not visual information. If visualizations appear at all in those approaches, they are tools for comparing abstract models rather than things which themselves are important to compare between versions.

Our attention on visual features overlaps with related work on the version management of UML diagrams, CAD documents, and other forms of visual modeling. Even in these areas however, computer-generated visualizations are favored over approaches that preserve visual information. Our approach to comparing visual differences may therefore add to those fields as well.

Since hypertext scholarship tends to focus on complex versioning models, solutions can seem intractable. Indeed, tasks such as comparing transclusion versions through time, or managing distributed version histories require nontrivial solutions. It might therefore seem that spatial hypertext, which adds complexity to the user interface, should require yet more complex version management.

Our approach to comparison simplifies the larger task of version management. Here, we propose an approach to computing and presenting differences between two collections[2] of spatial hypertext in a clear and useful way. By focusing on this single component, we identify and satisfy key requirements of spatial hypertext version management while clarifying some of the complexities associated with hypertext versioning in general.

2. THEORETICAL OVERVIEW

The calculation and presentation of comparisons is a fundamental component to any hypertext versioning or collaboration solution. Since different kinds of comparison take different roles in various versioning approaches, it is important to fit our definition in the broader context of version management before illustrating our approach with practical examples.

2.1 What We Mean by Comparison

For our purposes, a comparison calculates and displays all differences between two collections of hypertext, including all differences of structure, data, and presentation. For example, if structure is presented as location on an outline, and something in the second collection occupies a different location than it does in the first, that difference is displayed. If things are also presented

[1] Our ideas result from in-progress planning of a collaboration server for the spatial hypertext tool Tinderbox; our assumptions about spatial hypertext follow its assumptions [3].

[2] "Collection" sometimes refers to things which are visually grouped. We use it to refer to an entire spatial hypertext.

HT'09, June 29–July 1, 2009, Torino, Italy.

on a spatial map, the comparison should also show differences between the spatial maps. These differences could include anything, from colours and lines to metadata and formatted text.

Furthermore, to be useful within version management and collaborative systems, a comparison must also group sequences of change implied by difference. Although difference does not need to be understood as change, comparisons sometimes help users of versioning systems to understand or approve changes. Users should therefore be shown those differences upon which another difference depends. In a tree, for example, something cannot be the child of something else that does not exist. A user who approves the existence of a child must also accept the existence of its parent. To be useful therefore, comparison should calculate and present dependencies.

For our purposes, we assume that the hypertext consists of discrete, identifiable things. This is not the case for all systems. Different views of identity are central philosophical and technical viewpoints in hypertext research, and constitute a fundamental discussion to versioning. Since cases of ambiguity over identity, such as transclusion, are less associated with spatial hypertext than text, we will not concern ourselves with them here.

Comparison does no more than differentiate between two collections of hypertext, calculating and presenting the differences and dependencies between these collections. It does not account for a history of more than two collections, nor does it attempt to merge more than two collections.

2.2 What Others Mean by Comparison

Other kinds of comparison for versioning and merging have attracted much attention in research within hypertext and elsewhere. Within *version control*, evaluation of differences is sometimes used to decide when a new version of something should be created. Comparison is also sometimes employed for *model merging*, when two modified versions of an original must be re-integrated into a single version once again. Our approach, which addresses a different problem, is much simpler and more general.

Version control systems, which record changes as they are made, sometimes use comparison to improve efficiency. To avoid a new version being created every time a change occurs, some differences are considered to be too insignificant to cause the creation of a new version. Østerbyte suggests, for example, that the addition of annotation links to nodes should not merit the creation of a new version [11]. Similarly, in research on CAD versioning, Ahmed and Navathe suggest the division of objects in a project into those which are invariant in a version and unversioned objects which can be changed without creating a new version [1]. Similarly, the spatial hypertext software VKB groups history by significant changes [7] [12].

Here, we do not concern ourselves with ongoing evaluations of which differences are important and which are not. Instead, our approach is designed to be applied after changes have been made. A general comparison approach might show differences between subsequent versions of the same collection, but it might also be employed to compare collections with independent histories. One might also compare two very different things to show their similarities, a task outside the scope of version management software.

As with our approach, *model merging* is applied after changes have been made to a collection of documents. However, the primary interest of research on model merging is not on comparison itself. Instead, it focuses on how to make the merged product of comparison fit a set of constraints.

Consider, for example, the efforts of several developers who have been working on a project in ways that diverge. Tools for model merging attempt to assist the creation of a Union or Merged Project which satisfies defined constraints, such as a library specification.

Model merging is outlined in a paper on UML by Alanen et al., who explain that merging involves bringing together two sets of project differences, constrained by the state of a third, original project [2, p9]. This is similar to the approach taken in siDiff [13], a tool which visually suggests UML structures for merging forked projects. In this area, calculating differences is only the first step. The resulting sets of differences must themselves be compared to produce a merged project.

We are attempting to solve a simpler problem. Instead of comparing sets of differences in relation to some other constraints, we compare and display a single set of differences between any two collections and no more. While such differences are sometimes considered in model merging research such as Brunet et al., their research does not address visual information [4].

Model merging is also used in the Visual Understanding Environment, a spatial hypertext tool produced by Tufts University [15]. Its merge tool will take any number of hypertext collections and produce an additional hybrid collection. In this hybrid collection, darker colours are applied to those things which occur in more than one collection, while unique things are given lighter colours.

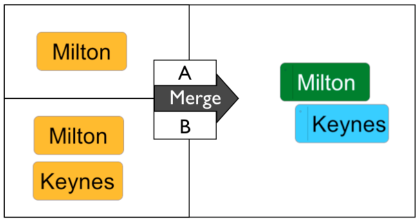

Figure 1: Merging Collections A and B into C in VUE

VUE's visually compelling merge tool is fundamentally different from comparison. Its merge operation differentiates between things found in multiple collections and things unique to a single collection. In doing so, it overwrites all differences with preference to the contents of a designated "primary" collection, avoiding comparison altogether.

2.3 Where Comparison is Not Appropriate

In addition to areas where spatial hypertext comparison fits into broader objectives and approaches to versioning, we have identified related questions which can safely be considered separate from the task of comparison. Chief among these is the association of the versioned information with timelines.

Systems that track all historical changes to something must do more than define what changes are significant. They must also decide on what things receive their own timeline, and where to place notches in that timeline. Nguyen et al. provide an example of this in their proposal of a system of product versioning for hypertext [10]. In their view, it is better to give an entire collection an overall timeline than to employ total versioning, where everything is associated with its own individual version history. This is the approach used in VKB [7].

If a generic spatial hypertext comparison tool must distinctly display each compared collection separately, as we argue from a series of examples which follow, it will not be possible to incorporate complex models of version timelines into the comparison tool directly. Instead, as Hicks et al. note, the selection of what to send to a comparison tool should be defined within the hypertext model itself [6, 3.7].

3. IMPLEMENTATION PRINCIPLES FOR SPATIAL HYPERTEXT COMPARISON

In the following section, we present a series of principles for comparing differences between two collections of spatial hypertext. The following examples demonstrate that comparison requires multiple kinds of views, that distinct views (as opposed to an overlay or composite view) are necessary, and that dependencies must be calculated and displayed.

3.1 Multiple Views Must be Made Available

Since spatial hypertext software can present visual information that is uniquely shown in one of many views, each of these views must be available to users when viewing differences. Consider, for example, changes to the font size in an outline which do not have a corollary in a spatial map:

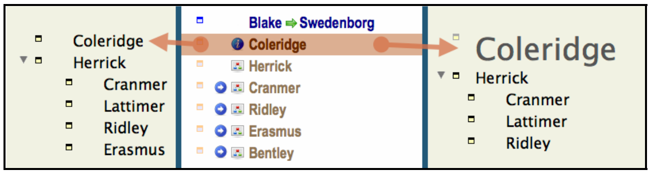

Figure 2: Difference in "Coleridge" outline font size

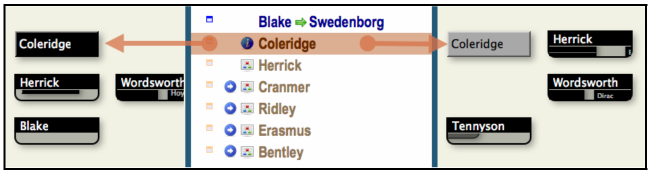

Figure 3: No difference in "Coleridge" map font size

While both views display a change in the "Coleridge" item from black to grey, only the outline view displays a change in the outline font size. This simple case demonstrates the need to present users with the option to choose multiple views when showing differences.

3.2 Collections Must be Displayed Distinctly

Recent implementations of spatial version management, most notably VUE, Webspiration, and siDIFF, present comparisons by overlaying a second collection on the first [15][16][13]. This approach however requires the software to distort or obscure important information from the comparison. We argue that these combined comparisons are inadequate, and that each collection must be displayed separately.

Overlaying both collections to compare them seems sensible; if something has moved or been introduced, such a change will be very evident. Overlay comparisons perform poorly however when things are removed or replaced. Unless visual features are modified to identify which things are from which collection, these differences may be imperceptible in an overlay.

While introducing new visual features into an overlay comparison removes some ambiguity, it introduces other problems. This can be demonstrated using VUE, which modifies colour when merging.

In a VUE merge, things which appear in both collections are given a dark colour, while things which appear in only one collection are given a light colour [Figure 1]. Both collection A and B include the item "Milton", while the "Keynes" item only appears in collection B. Modifying the visual properties, a necessary requirement of a combined view, has created ambiguity by introducing a colour difference not present in either collection.

3.3 Position Differences in Spatial Maps

Due to the nature of implicit structure in spatial hypertext [8], distinct views are also the best way to compare the position differences of things in a spatial map. While there is an intuitive attraction to showing 'before' and 'after' positions, perhaps with an arrow connecting these positions, such a display would be meaningless.

Our first example shows a scenario where an overlay of movement might seem useful: the item "Herrick" in the second collection is above and to the right of where it appears in the first:

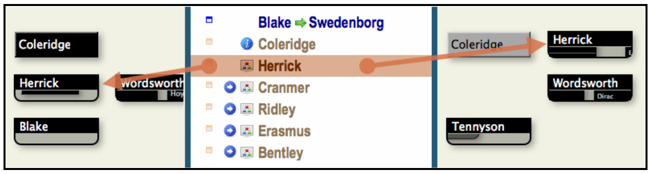

Figure 4: Comparing positions of "Herrick"

Because "Herrick" is the only thing to occupy a different position in the second collection, an overlay of movement seems a reasonable way to present the difference. For example, comparison software might show a faded "Herrick" in the first position and a fully-opaque "Herrick" in the second, with an arrow between them. Such a view would be ambiguous. It could easily be indistinguishable from a second, very different case: Instead of a moved "Herrick", the second collection could contain a faded copy of "Herrick" in a new position with a link to the original.

Furthermore, consider what happens when more than one thing occupies a different position:

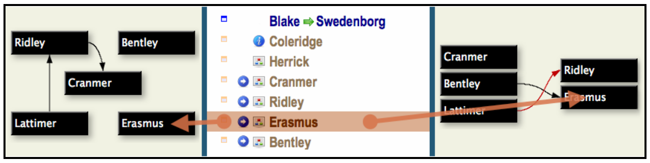

Figure 5: Comparing implicit structure for "Erasmus"

The idea of movement cannot explain the difference in Figure 5. Position in spatial hypertext implies meaning through spatial relationships. In Figure 4, we might permit the idea of movement, since most of the spatial relationships appear in both collections. But in Figure 5, the nature of the spatial relationships is very different in the two collections. A group with something in the centre becomes two vertical lists.

An overlay comparison might display "Erasmus" in 'before' and 'after' positions against one of the contexts. This would distort the meaning implied by a change in position. Alternatively, both contexts might be displayed in the same space. With so many differences, the composite would be unreadable.

Because overlays create unintended ambiguity at best, are unable to present the implied meanings important to spatial hypertext, and can become unreadable, they are not adequate for comparing spatial maps. Instead, the compared maps should be displayed in distinct views, as in the examples above, which preserve important visual details and context.

3.4 Link Differences

Both multiple views and comparison across distinct areas are necessary to properly display link differences.

Tools which permit several spatial areas per collection cannot display cross-area links using a single spatial map. A comparison tool which shows link differences should therefore present them in some additional way to spatial maps, such as a list.

As with position, the implicit relationships of links to each other and to spatial context require comparison across distinct areas. In the following example, the "Ridley" in the second collection has only one link, which is now red:

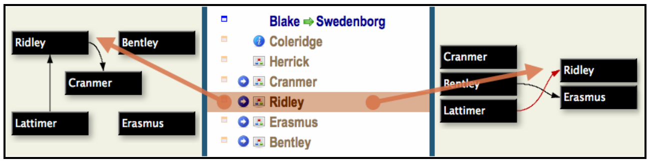

Figure 6: Comparing links associated with "Ridley"

In the first collection, the directional links from "Lattimer" to "Ridley", and from "Ridley" to "Cranmer" could imply sequence, or some other grouping of these three. In the second collection, a link from "Lattimer" to "Ridley" is also present, albeit in a very different spatial context. Here, it is horizontal instead of vertical and seems to imply a relationship between two vertical lists.

Simply listing link differences is insufficient because changes in vertex spatial positions can result in different implied meanings for a link. Spatial views are also insufficient; they cannot show all links. Link differences therefore require comparison using multiple, distinct views.

3.5 Dependencies for Outline Differences

Comparison must also calculate and present dependencies when displaying differences in collections which associate spatial areas in a tree structure, as is the case with Tinderbox. Furthermore, these differences must be compared with a view which shows the tree. No spatial view is able to display them; if something moves to a different tree location, it disappears from its former space.

Since version management software employs comparison to identify changes, and since rules about structure make some changes dependent on others, a useful comparison tool should present dependencies in addition to showing differences.

In the following example, "Swedenborg" can only be the parent of "Blake" if it is not Blake's child:

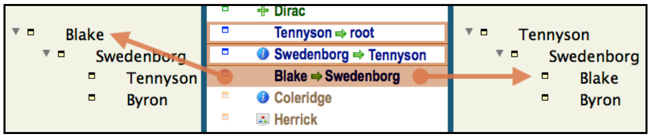

Figure 7: Dependencies required by parent inversion

"Blake" may only move from being a child of root to being a child of "Swedenborg" if the two outlined structural changes also take place. Thus, when the "Blake ► Swedenborg" move is selected, the other two moves should be highlighted as well. In contrast, the movement of "Tennyson" to root does not require either of these structural changes.

3.6 Identifying Difference

Elsewhere, we argue that visual identifiers of difference are especially ambiguous in spatial hypertext, since they are not distinguishable from things in the collection. Yet our demonstration examples use a red arrow to identify a specific difference. Here, ambiguity is prevented by the use of distinct views for presenting compared collections. Our identifying arrows cross view boundaries and cannot be confused with visual features of the collection to which they point.

4. COMPARISON IN HYPERTEXT MODELS

The approach we have outlined can supply an useful, independent component to existing hypertext version management systems. In the following section, we evaluate the potential role of comparison in several well-known systems.

4.1 WebDAV

Neither the calculation nor the presentation of comparison is an essential component of hypertext version management, as WebDAV illustrates. Currently the most refined hypertext version management system available, WebDAV is a box, not a display cabinet; it avoids presentation altogether. It cannot compare hypertext collections because it assumes nothing about the nature of the data it contains.

Any proper version management solution based on WebDAV would have to find comparison functionality elsewhere [17]. For HTML, which seems the most common hypertext model in WebDAV, this functionality can be provided by HTML-aware diff software. Likewise, any spatial hypertext models using WebDAV could employ an independent spatial hypertext comparison tool.

4.2 XanaduSpace

While XanaduSpace spatially orients hypertext collections, its key features exceed the limitations of our approach. Since XanaduSpace collections are presented in three dimensions, with positions calculated by algorithm, and since its text is fundamentally transclusive, our approach to comparing two-dimensional spatial hypertext does not apply [9]. It is probable, however, that many principles inherent to two-dimensional comparison also hold for three-dimensional arrangements.

4.3 Spatial Hypertext Applications

Current spatial hypertext applications can make use of comparison in three ways: comparing files from single-user applications; comparing collections in an online collaboration tool; and comparing versions in a version history.

Most spatial hypertext applications, such as VKB, Inspiration, Visio, VUE, and Tinderbox, are single-user applications which save collections to files. These applications have the greatest need for comparison, since collaboration and version tracking are poorly supported, and comparison is a crucial component of these tasks.

Recent alpha-quality projects, such as Webspiration and ShyWiki, permit multiple users to collaborate online [16][14]. Yet neither software provides a comparison facility. Without one, it is difficult to report and compare the contributions of different users.

Some applications, such as VKB and Webspiration, do track version histories. Both tools however can only display the state of the spatial hypertext at a single moment in history. Since comparison is how we make sense of changes over time, this strongly limits the value of recording version history.

5. DIRECTIONS FOR FUTURE RESEARCH

We have deliberately taken a narrow focus on comparison in spatial hypertext to solve a tractable problem and provide a starting-point for further research. Many interesting questions remain:

5.1 Merging Spatial Hypertext Collections

Since context supplies implicit meaning in spatial hypertext, approving changes selectively can result in meanings very foreign to either compared collection. Merge software will need special editing features to support accepting and merging changes.

Furthermore, implicit spatial meaning is often personal. Multi-user spatial hypertext may need to support comparison and merging in regard to an individual user's personal arrangement of a shared spatial hypertext collection.

5.2 Text and Transclusions on the Spatial Map

Although some spatial hypertext software displays bodies of text on the spatial map, we have not considered the issues involved in comparing them. Furthermore, if transclusions are permitted in the spatial map, comparison becomes more complex in ways not considered here.

5.3 Transclusions and Historical Links

By limiting comparison to distinct views of self-contained collections, we have been able to eliminate considerable ambiguity from how comparison works. According to Nguyen et al, modeling links in-between versions is "inconvenient" [10]. We suspect that comparing collections with these features will be difficult to achieve without introducing new ambiguities.

6. CONCLUSIONS

Comparison is an important but hitherto unexamined part of hypertext versioning and collaboration. Especially in spatial hypertext, whose uses tend naturally toward collaboration, comparison and merging are not well understood. The problems seem complex due to the complexity of different versioning approaches. Furthermore, the display of multiple versions, user contributions, and merges can distort important features of the spatial hypertext collection.

By producing a simple, general approach to comparison, we have been able to illustrate necessary principles for handling the challenges unique to comparing spatial hypertext. Multiple view options, distinct areas for different collections, and dependency lists are all necessary if comparison is to retain the kinds of meaning fundamentally important to spatial hypertext.

7. REFERENCES

[1] Ahmed R. and Navathe S. B., 1989. Version Control and Management in Computer-Aided Design Databases, Technical Report, CIS-TR-89-9, University of Florida, Gainesville FL

[2] Alanen M. and Porres I., 2003. Difference and union of models. In Perdita Stevens, Jon Whittle, and Grady Booch, editors, UML 2003 - The Unified Modeling Language, volume 2863 of Lecture Notes in Computer Science, pages 2–17. Springer-Verlag

[3] Bernstein M. Collage, Composites, Construction. In Proceedings of the 14th ACM Conference on Hypertext and Hypermedia (Hypertext 2003) (Nottingham UK Aug 26-30)

[4] Brunet G., Chechik M., Easterbrook S., Nejan S., Niu N., Sabetzadeh M., A Manifesto for Model Merging. 1st International Workshop pn Global Integrated Model Management (ICSE 2006 Workshop) (Shanghai, 20-28 May 2006)

[5] Diffutils: GNU. Accessed 04 Feb 2009: http://www.Gnu.org/software/diffutils/

[6] Hicks D., Leggett J., Nürnberg J., Schnase J., A Hypermedia Version Control Framework. ACM Transactions on Information Systems, Vol 16, No 2, April 1998, Pages 127-160

[7] Kim D., Shipman F. Efficient Branched History Management Based on History Segmentation. 2006 Symposium on Interactive Visual Information Collections and Activity (IVICA)(College Station, Texas, 20-27 Oct 2007)

[8] Marshall C.C., Shipman F. M. Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext. Proceedings of the Fourth ACM Conference on Hypertext (Hypertext 2003) (Nov. 14-18, 1993), pp. 217-230

[9] Nelson T. Back to the Future, Hypertext the Way It Used to Be, Proceedings of the 18th ACM Conference on Hypertext and Hypermedia (Hypertext 2007) (Manchester, UK, 10-12 Sept 2007)

[10] Nguyen T., Thao C., Munson E. 2005. On Product Versioning in Hypertext. Proceedings of the 12th international workshop on Software configuration management.(Lisbon, Portugal)(113 – 132)

[11] Østerbyte K. 1992. Structural and cognitive problems in providing version control for hypertext. In Proceedings of the European Conference on Hypertext (ECHT '92) (Milan,Italy, Nov. 30–Dec. 4), D. Lucarella, J. Nanard, M. Nanard, and P. Paolini, Eds. ACM Press, New York, NY, 33–42

[12] Shipman F., Workspace History and the Structure of Hypertext Activity. Second Workshop on Spatial Hypertext (College Park, Maryland, 12 June 2002)

[13] siDiff: University of Siegen, Germany: Accessed 18 December 2008. http://pi.informatik.uni-siegen.de

[14] Solis C., ShyWiki- A Spatial Hypertext Wiki. The International Symposium on Wikis (WikiSym)(Porto, Portugal, 08-10 September 2008)

[15] Visual Understanding Environment. Tufts University: Accessed 04 Feb 2009. http://vue.tufts.edu/

[16] Webspiration. Inspiration Software, Inc.: Accessed 04 Feb 2009. http://www.mywebspiration.com

[17] Whitehead J.. 1998. Collaborative Authoring on the Web: Introducing WebDAV. ASIS Bulletin (Oct, 1998), Accessed 4 Feb 2009. http://www.asis.org/Bulletin/Oct-98/webdav.html