A Scholia-based Document Model for Commons-based Peer Production

Joseph Corneli and Aaron Krowne

Abstract: Commons-based peer production is a term that describes authorship of shared information resources. In this article we examine the technical aspects of writing-in-common. We begin with a simple model: that of text and commentary. This scholia-based model emphasizes *ownership of speech* and *freedom of speech*. We then consider what happens when the *freedom to create derivative versions* is added to the mix. The resulting model proves to be quite sophisticated, and flexible enough to describe many different commons-based peer production systems. We provide an overview of our implementation of this model, and suggest some ideas for subsequent work. We conclude by discussing the implications of our model for distributed authorship and writing.

INTRODUCTION

The simplest model of a document is a list of characters; a string, file, or buffer. This model is not diachronic; a diachronic model keeps track of editing operations, or summarizes them as a sequence of differences between document versions.

Many texts have features which can not be modeled adequately without still more information: markup or metadata. Compilations, hypertext, and collaboratively written documents are examples. In this essay, we advance the idea that it can be useful to treat such documents and document ensembles as collections of *scholia*.

The Digital Library and the Document

For our purposes, the document and the library are essentially the same. In other words, the traditional library-document dichotomy can be viewed as a smooth spectrum, which we consider as a whole. Towards one end of the spectrum, the number of authors decreases and the topics under discussion become more integrated, and the information artifacts look more document-like. Towards the other end, the number of authors grows and the semantic gaps

M. Halbert (Ed.): *Free Culture and the Digital Library Symposium Proceedings*. Atlanta, Georgia: MetaScholar Initiative at Emory University, 2005. pp. 241-254.

between topics increase, and the information artifacts become more library-like. To be clear, when we wish to refer to elements of a given information artifact in a generic way, we will call them *articles*. We will use the term "scholium" to describe an article that is about another article. While a given article need not be about another article in general, our view is that it is about *something*, or perhaps about many things. An article that does not express these relationships explicitly is a degenerate scholium.

Scholia

According to Webster,[1] a *scholium* (pluralized *scholia*) is,

1. A marginal annotation; an explanatory remark or comment; specifically, an explanatory comment on the text of a classic author by an early grammarian;

2. A remark or observation subjoined to a demonstration or a train of reasoning.

The Talmud is an excellent example of a document that is comprised of scholia. This document is a collection of interconnected commentaries and reflections on Jewish law (the *Torah*), composed by generations of Jewish religious scholars. There is an obvious and oftnoted comparison between the internet and the Talmud: we readily see ancient scholia-based documents as primitive hypertext. However, a deeper comparison is not simply technological but psycho-social.[2] There are similarities between our experiences of "culturally comprehensive" documents, whether secular or religious in nature. Of course, if you subscribe to the McLuhan view of media ("the medium is the message"), you would expect to see similarities (Federman 2004; McLuhan 1964).

Scholia also appear in the writings of mathematical scientists (*e.g.* Euclid, Galileo, and Newton) or philosophers writing in a similar style (*e.g.* Spinoza). Today, mathematicians typically call peripheral observations (frequently, of secondary importance) "corollaries" or "remarks." Mathematicians also mark up parts of their texts as "axioms," "definitions," "propositions," "theorems," "lemmas," or "proofs." Markup, in general, can be thought of as scholia-based, *i.e.*, as commentary that instructs the typesetter or the reader to treat a given piece of content in a certain way.

Free culture and the scholia-based document

This paper, and the system it describes, are a response to certain concerns that face us when we work with and think about shared information resources. These concerns have to do with the issues of *intersubjectivity*, *ownership*, and *freedom*. Content in a shared document is typically of non-trivial intersubjective importance. However, it is more than likely that different parties will have some different ideas about the information that is presented in a given text (it is no coincidence that the root of *intersubjective* is *subjective*)!

Free content (*i.e.* free as in freedom, or *libre* content) is one way to nurture difference. Free content can be modified and redistributed (and, in particular, *forked*) without permission or apology. Nevertheless, free content typically manifests aspects of a common resource as well as an open access resource; while anyone can do essentially whatever they wish with the content offline, in its online life, the content is managed in a socially-mediated way. In particular, rights to *in situ* modification tend to be strictly controlled. The details differ from ownership model to ownership model (*e.g.*, PlanetMath has article owners and access control lists, software projects like Emacs have a limited number of developers with commit privileges, members of the Wikipedia community enforce rules about what kinds of content are allowed, and so forth). In these instances and generally, unwillingness to cooperate comes at a cost of valuable support from the community, which must be balanced against the limitations the community imposes on the individual.

By finding new ways to support freedom of speech within CBPP documents, we embrace subjectivity as a way to enhance the content of an intersubjectively valued corpus. In the context of "hackable" media and maintenance protocols, the semantics with which scholia are handled can be improved upon indefinitely on a user-by-user basis and a resource-wide basis. This is free culture in action.

Hyperreal Texts

Our interest in scholia-based documents largely derives from an interest in helping assemble a human-friendly, AI-enriched mathematics learning and communication interface. The immediate goal is a system in which text and code are both well-supported, as two sides of the same coin. Qualitatively, we want want to be able to represent and work with complex ontological

relationships between entities that are encoded in the system. The system should be useful for both humans and programs—balance along the human/artificial axis is as important toward this end as balance along the freedom/ownership axis or the intersubjective/subjective axis. In short, we find it natural to use a scholia-based platform as the foundation of our *Hyperreal Dictionary of Mathematics* project.[3] Applications in other areas of intellectual inquiry (including digital libraries), as well as in business, government, and day-to-day life also seem to hold great promise.

Inspiration

We have already seen glimpses of the ideology informing this paper (in particular, we've touched on free software, commons-based peer production, the Talmud, and mathematics). We'll come to more of this later on. Here, we would like to describe certain key technological inspirations. These should serve to illustrate the naturality of a scholiabased document model, as well as to contextualize our implementation of the model. Usenet, Slashdot, the World Wide Web and WikiWikiWeb are all inspirations. In the first two, the most obvious type of scholium is the *followup*. In the latter two, it is the *link*.

PlanetMath[4] uses several types of scholia. Discussion fora are attached to pretty much every "substantial" object in the system; there are auto-generated invocation links between articles; and attachment relationships can be asserted to apply to between objects. Objects can themselves be distinguished as being one of several different types and belonging to one of a number of different subject categories. Also, and most importantly for the current discussion, PlanetMath employs an explicit article ownership model.

Emacs text properties provide a facility for unlimited markup of strings, buffers, and files.[5] However, text properties are hierarchical (treelike), whereas we are looking for something web-like. Of course, locally, text properties are great (and, indeed, we use them in our implementation).

The semantic web project is exploring ways in which to make metadata about web pages available to computers with reasoning capabilities.[6] Our scholia-based documents are small "semantic webs" with a particular flavor, though it may be more appropriate to use the older and more generic term "semantic network" (Quillian 1968) to describe them. As a data structure, a scholia-

based document is a network with arbitrary text (and metadata) on its nodes and arbitrary annotations on the connections between nodes. Graphically, this closely resembles the notion of a *concept map* (Novak 1998), albeit with significant text content attached to each node.

Semantic networks and concept maps are just two of the many different strategies for knowledge representation. Cyc[7] and KM[8] are two other systems we've studied. Our goal is a system with similar expressive power, but which is more human-friendly (with *reasoning* capabilities to come eventually).

The first author's earlier hypertext experiments were heavily inspired by LISP, and the results ended up resembling Gopher (Karger 2000). The current system is meant to be much richer, but it is still LISP-inspired, particularly in terms of support for self-modification. Indeed, one of the key inspirations for the current system are the text-based "game" environments (MUDs,[9] etc.), in which people are able to "hack" the text-based virtual worlds that they are interacting in and with, in real time.[10] Scholia can include actionable features, and with appropriate support, contributing authors can modify the medium in which the scholia reside, not just its content.

Survey

The scholium-based document is not a new idea. Here, we are referring not to classical works, but to Ted Nelson's project and document model entitled *Xanadu*, which is described in his book *Literary Machines* (Nelson 1990), originally published in 1982. His development strategy was quite different from the one taken here, however—in particular, while he and his cohorts were focused on developing powerful server-side technology, we focus on implementing user-ware with a simple p2p extension. Importantly, while at least some of Nelson's ideas have been implemented and released under a free software license, the code didn't compile for us.[11] A biography of sorts together with a history of the Xanadu project has been published in *Wired*.[12] Nelson is admired by FSF General Counsel, Eben Moglen, for "identifying the predicament of information ownership in the digital age" (Williams 2002).

One rather important difference between Nelson's vision (as expressed in *Literary Machines*) and the present one is that he deprioritizes artificial intelligence in favor of human intelligence. As remarked above, actionable features and artificial intelligence

occupy an important place in the scholium system. Our situation with regard to artificial intelligence is similar to the one described by Minsky in *The Society of Mind* (Minsky 1985). The connection between ideas and agents is not so dissimilar; only the translation to actionable form is missing to make an idea into an agent. Meaning clusters translate to complex agencies. As time goes by, we expect to find actionable and non-actionable features paralleling one another along some dimensions, intertwining along some, and diverging on others.

"Superimposed information" is a subject of current research in the field of digital libraries.[13] Note that while this model is *locally* similar to the scholia-based document model, this branch of research focuses on one superimposed layer. This makes a certain amount of sense for traditional libraries, which hold a specific, administratively-controlled collection of information. The superimposed information model emphasizes making this primary artifact more useful via value-added "attachments" (annotations and so forth). However, for us, neither library nor document is static, and annotations are an integral part of both. Thus, scholia-based documents are as much "community" as they are "collection."

We find it compelling that, in the context of a digital library, marginal conversations within a text provide a chance for readers to interact with primary authors and with each other, and to become primary authors themselves, all at once. While marginalia are considered to be *vandalism* in physical library books, in a digital library, there is no reason to fear them—they can easily be hidden away. The scholia-based document model reflects the standard postmodern pun which says that writing on a text or subject (*i.e.* criticism, discourse) is writing *on* the text (making ones mark). We expect meaning to accumulate in the "margins" of texts, and for meaning complexes to grow by stitching documents together along their margins.

IMPLEMENTATION OVERVIEW

In this section we give a tour of our implementation of the scholium system. The critical elements of the system are described in general terms here. Details, including code (and literate markup of the code) may be found at (Corneli 2004).

Articles

The foundation of the scholium system is a catalog of articles (list or hash table). Adding or updating the reference for an article to the catalog is the fundamental operation in the system. This operation stores

- the article's name;
- its text or a pointer to its text;
- a record of what it is about—nothing if it is degenerate, otherwise, some article(s) or passage(s) of articles in the collection;
- the designation of a type (link, followup, forum, action to take if a certain event happens at the parent, etc.);
- and bookkeeping information to keep track of ownership information and editing history.

In Figure 1, we illustrate the conceptual model of the scholium system with a hypothetical content instance. Shown are the key entities of *people*, *articles*, and *references*.

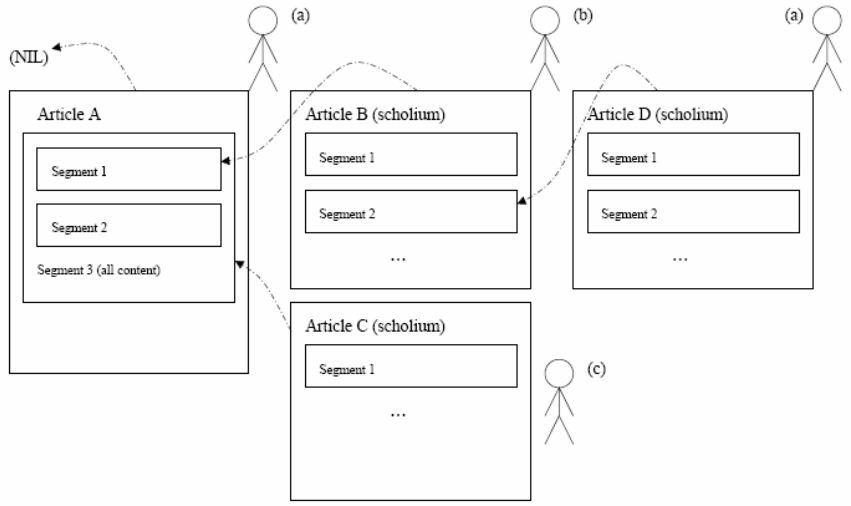

Figure 1: The key elements of the scholium system, for a hypothetical small document/library. There are three people (a, b, and c), and four articles (A, B, C, and D). All articles are scholia (i.e. refer to another article), except A. The contents of each article have been broken into "segments" for clarity. Links are shown as dashed arrows from articles to article segments.

Environment

The current prototype system runs under GNU Emacs. We try to conform to the Emacs Way to the greatest extent possible. Typically, scholia are written about files and buffers, and are displayed alongside them. The default display uses color to associate displayed scholia with regions that are being commented on, but delimiters can be used for this purpose as well (for the benefit of those working without fontlock). Color can also be used to distinguish between different types of scholia (*e.g.* comments by different authors).

Our ability to render articles is limited to things Emacs can display: plain text, code, and pictures all work, but special proprietary formats are not supported in this implementation, nor do we have support for making scholia about specific pieces of a rendered diagram, for example.

The user can navigate the display in various ways, e.g., by scrolling between marked-up regions of the main article, or finding the region(s) associated with a given scholium, or the scholia associated with a given region. Additional browsing methods are described below.

Displaying Scholia

When an article is displayed, the system finds attached scholia and displays them too. Selective displays are possible (*e.g.* just show all the *links* attached to a given article). Finding scholia in the catalog requires search; sometimes we can limit the search to make it faster (this will be described further below). In Figure 2, we show a number of ways in which the underlying content from Figure 1 can be displayed.

Adding to the library

We support various convenient ways to add articles. Different kinds of articles need different treatment. For example, the content of a buffer is lost if the buffer is killed, so a backup of the buffer's contents should be made immediately. In addition to specifying the article's text and stating what (if anything) it is about, one can specify the article's type. We provide built-in functions for creating scholia about the current buffer, and for creating scholia about mixed collections of other articles and passages.

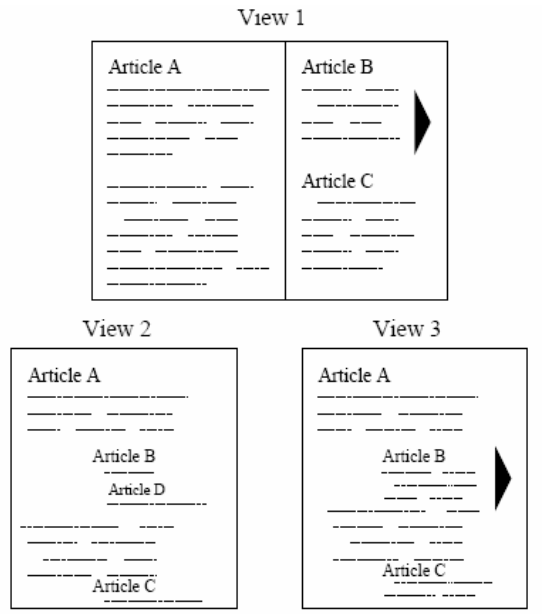

Figure 2: Different possible views of the articles from Figure 1. Three views are shown. View 1 is a two-pane style, with attached scholia on the right and the main article on the left. View 2 includes only title and linkage information for attached scholia, unfolding recursively to a user-specified depth. View 3 is a single-frame view, with one level of attached scholia inclusion (deeper levels are indicated with an arrow).

Browsing

We provide several different browsing mechanisms. Simple local navigation features (scrolling etc.) were mentioned above. “Following” a scholium to make it into the new current article is supported. Browsing by catalog (*i.e.*, by list) is supported; we use a generic menu mechanism that make it easy to select different collections of articles matching different criteria, and perform various actions on them. Finally, web-browser-like history-based navigation is provided.

Editing, deleting, saving, restoring

Articles are versioned and entire document versions or, alternatively, a collection of patches sufficient to move between versions, is maintained for each article. When articles change, it is typically necessary to propagate data to attached scholia in order to maintain coherence of references. Propagated changes need not be accepted, in which case the attachment relationship is fixed at the previous version.

Similarly, permission is typically requested whenever a portion of an article that has been *transcluded* changes. ("Transclusion" is Nelson's term for "included by reference"; both transclusion and inclusion of pieces of other articles are supported.)

Changes that take place to articles outside of the system (*e.g.* moving a file from the shell) are of course nearly impossible to deal with. However, scholia can be saved in explicit, invariant forms. (CVS is natively supported).[14] Saved articles can be read back in selectively.

Currently, name conflicts pose a bit of a problem; if every scholium has a unique name, then the problem goes away. In general, we can approximate a solution by asking the user to uniquify names when conflicts are encountered.

Namespaces

Another solution to the problem of name conflicts is provided by namespaces. For example, if article *A* has type "sublibrary" and we read in a scholium of type "member of sublibrary *A*," then we don't have to worry about name conflicts with articles not in sublibrary *A*. Namespaces can be used to make search convenient. For example, we can store all assertions of type *Z* in a given namespace; then if we are building a display that only relates to objects of type *Z*. In other words, we need only search one namespace instead of the whole library.

DISTRIBUTED AUTHORSHIP

Distributed authorship is actually very easy using this system. Each contributing author posts her or his contributions to the document at their own chosen location. Then the other authors download the articles stored in all of the locations they want to use. (All authors don't necessarily need to have all of the articles.) In the case of document revision, changed versions are simply posted, and other authors learn of the changes whenever they sync.

Derivative versions

In general, the author may be the only person with permission to create modified versions, but we can also distribute this permission over a wider group of people. As we've already mentioned, derivative versions can be put together using two principles, inclusion and transclusion. Inclusion tends to reduce search but increase storage.

Here we illustrate *tracking* of derivative versions with a scenario. Suppose that a series of definitions was quoted in a textbook-style entry. The author of the textbook might receive a question from a reader and then adjust one of these definitions to include more expository text. If the original author was tracking derivative versions, the new expository text could be added as a scholium attached directly to the original, or the original could be modified.

These sorts of exchanges should still be possible even if one of the agents is working outside of the scholia-based system. In order to make *bi-directional updating* work in this case, both content-sharing parties need to be able to read a stream of diffs generated by edits taking place in another system and decide how to incorporate the modifications.

These issues should be of interest to anyone maintaining a collaborative digital library; information-sharing between such entities typically needs to support content that can change on both ends of the pipeline.

FUTURE WORK

Some ideas for explorations to undertake with the scholium system and related concepts follow. Note that some of these could be considered feasibility or proof-of-utility experiments:

- Import a wiki and build a wiki-like interface to the scholium system;
- Use the scholium system to write a synchronous or asynchronous multiplayer game;
- Use the scholium system to maintain a text-based forum or set of fora, as found on PlanetMath or Slashdot;
- Implement a Slashdot-like scoring system for quality control;
- Use the scholium system to manage an evolving codebase (*i.e.*, take advantage of the functionality which subsumes a system like CVS);
- Use the scholium system to manage TODO lists;
- Port the GNU Collaborative Dictionary of English (GCIDE) to Emacs and give instant access to definitions as scholia;
- Implement code to make an index or do autolinking as you type;
- Port WordNet15 to the scholium system and use the system to collaboratively improve the database;

- Implement semi-automated content sharing between two collaborative digital libraries, for instance, PlanetMath, and Wikipedia.

This list should give some idea of the range of capabilities the scholium system, in theory, encompasses. Long term investigations under conditions of fairly wide uptake would shed light on broader social implications of the system. We imagine that social institutions—from peer review to popular science, and from online shopping to participatory government—would tend to be transformed by widespread use of these systems. In fact, some of this social transformation is already apparent—in specialized implementations of the scholia concept (*e.g.* weblogs, wikis, online forums, PlanetMath). Currently we can only speculate as to how the general system outlined here would interact with these trends.

One thing we can say at this point is that the model seems to provide a useful basis from which to explore the design and implications of social contracts in online communities. We hope future work will take up this issue.

CONCLUSION

In this article, we have described a scholia-based document model and outlined an implementation of a system that supports this model. We have discussed some ramifications of scholia-based documents and libraries. We have have also shown how the model can be used to facilitate powerful collaboration dynamics in a wide array of scenarios and social settings—by fostering and managing alternative perspectives, encouraging responsible maintainership, and enabling readers to routinely make useful contributions.

The scholia-based document model corresponds to a culture with empowering conceptualizations of freedom and ownership. Accordingly, this paper has been a description of a model as well as something of a social manifesto. We hope to see the ideas presented here take off, as we and others work to push the limits of the model.

Acknowledgments

Thanks to Ray Puzio for helpful comments and encouragement as the scholium system developed. Thanks also to Thien-Thi Nguyen and Sacha Chua for their comments on free software and hypertext, including comments on an earlier prototype of the system by the first author.

ENDNOTES

1. http://www.ibiblio.org/webster/.
2. See “The Talmud and the Internet” (Rosen, 2000). You can also find the Talmud on the internet, at http://www.sacred-texts.com/jud/talmud.htm.
3. See http://planetx.cc.vt.edu/AsteroidMeta/HDM.
4. http://planetmath.org/.
5. (info "elisp(Text Properties)")
6. http://www.w3.org/DesignIssues/Semantic.html.
7. http://www.cyc.com/cyc/technology/whatiscyc.
8. http://www.cs.utexas.edu/users/mfkb/RKF/km.html.
9. Marshal McLuhan’s theory ties in very nicely with MUD-like systems. Those with a user-hackable infrastructure embody a distinctly free “message.” For cultural materialists, the upshot is hackable superstructure. The fact that the MUD is virtual to begin with adds an interesting twist in this analysis.
10. It is interesting to compare the experience of these immersive worlds to the experience of internet or Talmudic scholars, mentioned previously. See (Rheingold 1993).
11. Xanadu development seems to be going slowly at present; see http://udanax.com/ and the mailing list at http://xanadu.com.au/mail/ .
12. See http://www.wired.com/wired/archive/3.06/xanadu_pr.html, but note that this work is not endorsed by Nelson, who writes at http://ted.hyperland.com/whatsay/ “I believe the piece is a study in cunning and deliberate dishonesty, the most dastardly piece of dirty journalism I have ever seen.”
13. For example, see (Maier and Delcambre 1999), http://nsdl.org/community/projects.php?this_sort=start_date&keyword=&project_id=0435496, or http://datalab.cs.pdx.edu/sparce/.
14. Code Versioning System, an extremely popular free software program to collaborative manage software codebases. See http://www.gnu.org/software/cvs/.
15. WordNet is a “lexical database for the English language”—essentially a semantic network of words and relationships, which can be used as a dictionary or thesaurus. See http://wordnet.princeton.edu/.

REFERENCES CITED

Joe A. Corneli. A scholium-based document model, 2004. URL http://planetx.cc.vt.edu/AsteroidMeta/A_scholium-based_document_model.

Mark Federman. What is the meaning of the medium is the message?, 2004. URL http://www.mcluhan.utoronto.ca/article_mediumisthemessage.htm.

Bjorn Karger. The gopher:// manifesto, 2000. URL gopher://gopher.quux.org/0/Software/Gopher/whygopher/gopher-manifesto.txt.

David Maier and Lois M. L. Delcambre. “Superimposed information for the internet.” In Sophie Cluet and Tova Milo, editors, *ACM SIGMOD Workshop on The Web and Databases*, 1999. URL http://www-rocq.inria.fr/~cluet/WEBDB/maier.pdf.

Marshall McLuhan. *Understanding Media: The Extensions of Man.* McGraw Hill, 1964.

Marvin Minsky. *The Society of Mind.* Simon and Schuster, 1985.

Theodore Holm Nelson. *Literary Machines.* Mindful Press, 1990.

Joseph D. Novak. Learning, *Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations.* Lawrence Erlbaum, Mahwah, NJ, 1998.

M.R. Quillian. “Semantic memory.” In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, MA, 1968.

Howard Rheingold. *The Virtual Community*, chapter 5. Perseus Books, 1993. URL http://www.rheingold.com/vc/book/.

Jonathan Rosen. *The Talmud and the Internet.* Farrar Straus Giroux, 2000.

Sam Williams. *Free as in Freedom: Richard Stallman’s Crusade for Free Software*, chapter 13. O’Reilly, 2002. URL http://www.faifzilla.org/.