

Takeda, T. & Suthers, D. (2002). Online Workspaces for Annotation and Discussion of Documents. To appear in Proceedings of the International Conference on Computers in Education, December 3-6, Auckland, New Zealand.

Online Workspaces for Annotation and Discussion of Documents

Toshiyuki Takeda
Center for Information and Media Studies
Kwansei Gakuin University
Uegahara 1-1-155
Nishinomiya, Hyogo, 6628501, JAPAN
takeda@kwansei.ac.jp

Daniel Suthers
Laboratory for Interactive Learning Technologies
Department of Information and Computer Sciences
University of Hawai'i at Manoa
1680 East West Road, POST 309
Honolulu, HI 96822, USA
suthers@hawaii.edu

Abstract

Pink is a system that supports threaded discussions about artifacts (such as source code or security bulletins) by making it easy for users to refer to and annotate parts of the artifacts. Pink supports multiple types of artifacts and annotation patterns while maintaining appropriate separation of content and view through a three-tier architecture. This system is implemented as server-side scripts and operates with any standard WWW browser.

1. Introduction

With the expansion of the Internet, system administrators cannot easily keep up with new viruses and security vulnerability information. As Code Red and the Nimda Worm spread globally in 2001, the understanding of how to prevent such viruses was not sufficiently disseminated from system administrators to users. Although companies like Microsoft and CERT published security information, many users did not take action. Myriad security advisories confused many users because the system environments they covered differed from those of the users. Here, the issue of context sharing comes into play, as many users cannot identify the context of the messages they read, or identify the portions of the messages that are relevant for their own context. Popular tools such as Email, Netnews and the WWW were created to facilitate the sharing of knowledge. However, these tools are sometimes less effective because they do not facilitate context sharing.

1.1. Artifact-Centered Discourse

We call the type of discussion and argumentation that should be supported in the applications just described Ar-

tifact Centered Discourse [9]. Others refer to these applications as Anchored Discourse [4] or Contextualized Discussion [11]. The fundamental requirement in these applications is that the connection between messages and their contextual artifacts be maintained.

We discriminate three types of Artifact Centered Discourse (ACD); Parallel ACD, Embedded ACD and Linked ACD.

In parallel ACD (Figure 1), discussion tools and shared artifacts are displayed on entirely different screens. There is no communication or coordination between the discourse and the artifacts; they are simply displayed in parallel. The advantage of parallel ACD is that a user can change interface as one wants. The disadvantage of it is that a user might lose track of which part of an artifact is being discussed.

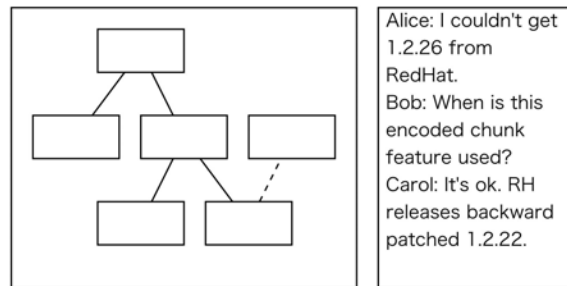


Figure 1. Parallel Artifact Centered Discourse

An embedded discourse representation embeds comments directly in the artifact under discussion, or achieves a similar effect by inserting comments for display without modifying the original artifact. Because the discourse always takes place in the context of the artifact, embedded ACD (Figure 2) has the advantages that it is easier to refer to parts of the artifact or to recover the portion of the

discussion that is concerned with a given part.

The fact that the discourse is distributed across the artifact in embedded ACD leads to one of its disadvantages: it is more difficult to get a sense of the whole discussion or to notice relevant relations between discussions about different parts of the artifact. The artifact may also become cluttered with comments.

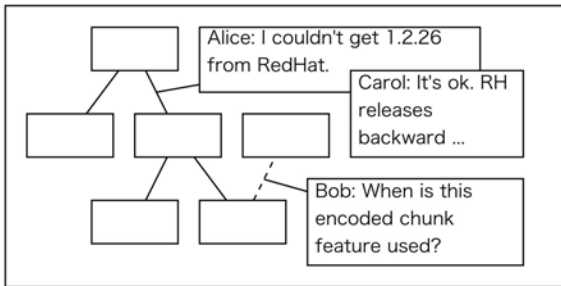


Figure 2. Embedded Artifact Centered Discourse

Linked ACD (Figure 3) can resolve the disadvantages of parallel and embedded ACD. In linked ACD, discourse representations and contextual artifacts are displayed side by side, as in parallel ACD, retaining the reply structure and chronology of the discourse. This approach resolves the tradeoff between the unrelated representations of parallel ACD tools and contextualized yet conversationally fragmented discourse contributions of embedded ACD tools. We believe that linked ACD, properly designed, can improve coherence and convergence of artifact-centered discourse by collecting together topically related contributions. The disadvantage of linked ACD is that it requires a larger screen.

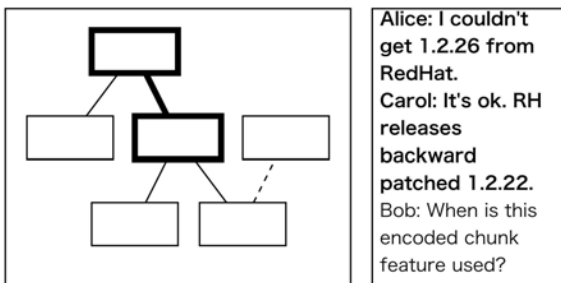


Figure 3. Linked Artifact Centered Discourse

This paper describes the software system Pink, which was created to support Artifact Centered Discourse. Pink supports the understanding and creation of artifacts that reflect intellectual discussions among participants. Since Pink

is client server software, it requires only a WWW browser for use.

1.2. Related Work

Previous research has shown that annotating text online enables participants to find relevant information more easily. The Annotation Engine [12], ComMentor [6], CoNote [2], CritLink Mediator [14], and the Journal of Interactive Media in Education [8] are web based online discussion systems with similar approaches. Microsoft SharePoint Team Services [7] is one of the out-of-the-box solutions. Kukakuka [10] is under development in the same laboratory as Pink and has many of the same objectives. In Kukakuka, a collection of Java servlets associates web pages with NNTP discussion groups and threads, presenting these together in a web client using frames. Many of the systems described above support only annotations to web pages, whereas Pink supports not only web page annotation but also embedded annotation to documents, such as Wiki-Wiki [5] documents, created within the system.

2. Software Design

Design issues for Pink system are as follows.

1. Artifact Centered Discourse support. The system supports Artifact Centered Discourse, especially linked ACD to discuss the contents of the artifacts. It is important to clarify and share assumptions, background knowledge and limits of applicability, which are not noted in the artifacts. Discussion in a shared context requires support of two functions:
 - (a) Artifact to Discussion reference. Enable users who are reading an artifact to access comments on a specific annotated region of an artifact. This allows users to share knowledge with someone who has the same interests.
 - (b) Discussion to Artifact reference. Enable users who are reading and responding to a threaded discussion about an artifact to access and refer to the relevant passages of that artifact. Since artifacts such as security advisories are quite large, this function should highlight the portion of the document that is being discussed.
 - (c) Building shared knowledge. Summarize and share as a new artifact the created knowledge that becomes clear to participants in a discussion.

2. Extensibility.

- (a) Multiple types of Artifact. Supports artifacts internal to the server (e.g., text documents and WikiWiki [5]), on external servers (e.g., web pages), or offline artifacts (e.g., citations to books or journal articles). The system has meta data with Artifact information.
- (b) Multiple annotation patterns. Proper annotation pattern varies with artifact pattern. For example, software source code should be annotated on line(s) or function(s), and a document should be annotated on sentence(s) or paragraph(s).
- (c) Flexible view. Content and its view should be separated and workspace manager can change to meet their needs.
- (d) Multiple repository platforms. A repository (database) of the system can change to meet their needs, such as performance or other administrative reasons.

3. Multiple client platforms. Enable users of a variety of hardware and software platforms to access the community workspaces (including artifacts and discussions). Objects in the system should be accessible from any standard based WWW browser without client side software, such as Java applet or Active X.

2.1. A Sample User Scenario

The next two sections show how users read and write in the workspace.

2.1.1 Reading and Annotating Artifacts

A user can begin either by reading a document artifact or by reading topic threads of interest. References are displayed as links in both directions. When the user is reading a document and encounters an annotation, the user can click on the annotations reference number in the artifact to read the related discussion. Conversely, a user who is reading a discussion can click on the numbered references to view the referenced artifact. (The referenced artifact may be the document itself, or a link or citation to the artifact in the case of external documents, respectively.)

A user creates a reference to a portion of a plain text or WikiWiki artifact as follows. The user shifts to a page for inserting a special tag when the user clicks a button in the artifact-browsing page. The user then inserts tags to indicate the extent of the reference, and clicks submit. A reference is then created in the repository and threaded discussion menu page.

Once a reference has been created, it shows up as a new thread in the discussion thread view. The user then adds

a Note to the Reference by clicking on [Write Comment], which loads a simple form for entering the Note.

2.1.2 Reading and Writing a Comment

The subject of the created Note is displayed in the discussion view and users can reply to the Notes as in a normal threaded discussion.

The Reference is shown as an anchor in the document with the number of the reference. When a user clicks this part, the corresponding part in the threaded discussion is shown and a user can read notes attached to the annotation point. Conversely, if a user clicks the number of a reference on a discussion thread page, the part of the artifact that includes the annotation point is shown.

2.2 Architecture

Pink has a Three-tier architecture (Figure 4) consisting of Presentation, Model and Repository.

Presentation. The Presentation layer generates html using objects in the model layer and parameters from a user. It uses two important objects: WebComponent to wrap model objects and WebTemplate for a template to generate html.

Model. A Model consists of a Workspace, and Artifact, Note and Reference objects derived from the Workspace. There are also objects related to users and access control.

Repository. The Repository is a persistent database in which to store objects. It is designed to use simple interfaces like a space-based repository, TupleSpace [3], so it is easy to implement a Repository in many ways.

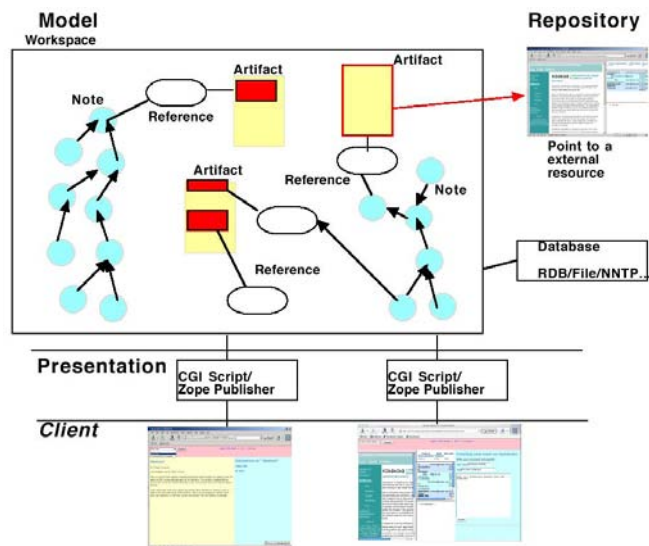


Figure 4. Architecture in Pink System

Workspace. The Workspace is a place to share and exchange knowledge by setting an annotation on an artifact and writing a note to a threaded discussion. A user can add or create an artifact, annotate it, and write a comment about the annotated part of an artifact.

Artifact. The Artifact object is used for representing a text document, web page, a book or journal article, etc. There are two kinds of Artifacts: structured and unstructured. The structured Artifact is composed of parts like W3C DOM [13] and an annotation point can be set on each part of an Artifact. (See Figure 5)

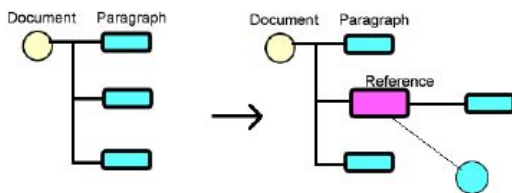


Figure 5. Reference Insertion

Reference. Reference is a pointer object that refers to either a part or whole of an artifact object and is often a root point of a discussion thread. *Note.* A Note object includes a user’s writings about something in a workspace and usually has a pointer to either a reference object or another note object.

2.3. Implementation

The current version of Pink is implemented on Zope (web application server and content management system written by Python programming language [15]). All object classes including Artifacts and discussion items and its containers (threads) are defined as Python objects. New Artifact types can be written in Python when needed. Zope supports many database management systems as repository with fine abstract interface so that Pink repository can be changed easily.

Figure 6 shows a sample user interface when a user make an annotation on a structured artifact.

3. Future Work

The current version of Pink allows the user to set an annotation point on an entire external web page. Annotation of parts requires a filtering object that takes in an external page and changes it into a model object. This is a kind of proxy server function, the way CoNote and CritLink Mediator work. Pink already has similar functionality, but there are two reasons not to deploy this function as a service now.

One is the problem of intellectual property rights. Rewriting a certain page and showing this revised page to a user may not be allowed in some cases, even though Pink just appends tags and doesn’t change any contents of an artifact. For the Internet community to share knowledge, permissible usage should include at least proper quotation.

Another reason is the problem of version management. The Web page used as a target artifact will be referenced by its structure, which is assumed to be constant. However, Internet documents on active sites do change, especially security advisories. The new version of the artifact may not include all the contents mentioned in the past discussion. When the system detects changes in an external web page or finds that a discussion thread is connected to obsolete content, the simplest approach would be to treat the old annotation and discussion as obsolete, and to only display non-obsolete and new annotations.

However, valuable knowledge may lie in these obsolete discussions. Consider for example a workspace recording design rationale for software revisions, or a learning application in which a student posts a document on a web server, an instructor comments on it, and the student then revises the document accordingly. The discussion for such frequently changed sites needs to include older versions. A better approach is needed. Ideally, the system would enable users to browse the differences between old and new artifacts, and provide access to the old discussion thread generated by the prior versions. On the other hand, there are no problems for the contents created in the Workspace because the system can show the part changed in the artifact easily. Another area for further work is to support structured discussions like IBIS [1]. The IBIS scheme of Issue, Position, and Argument is used to record argumentation while exploring wicked problems. Since our approach focuses on the relationship between artifacts and discourse, further examination is required to introduce discourse-structuring devices.

4. Summary and Conclusions

The Web has fostered explosive growth in a variety of online communities. Many of these communities would benefit from better tools for online discussions that are focused on the interpretation and/or creation of shared documents and other artifacts. Functional requirements of tools for artifact-centered discourse include the ability to move between discourse and artifacts in both directions: retrieving a discussion associated with an artifact or portion thereof; and bringing up an artifact that is referenced by a contribution to the discussion.

This paper described the architecture and interface of Pink, a software system that meets these functional requirements. The system is based on an abstract 3-tier server ar-

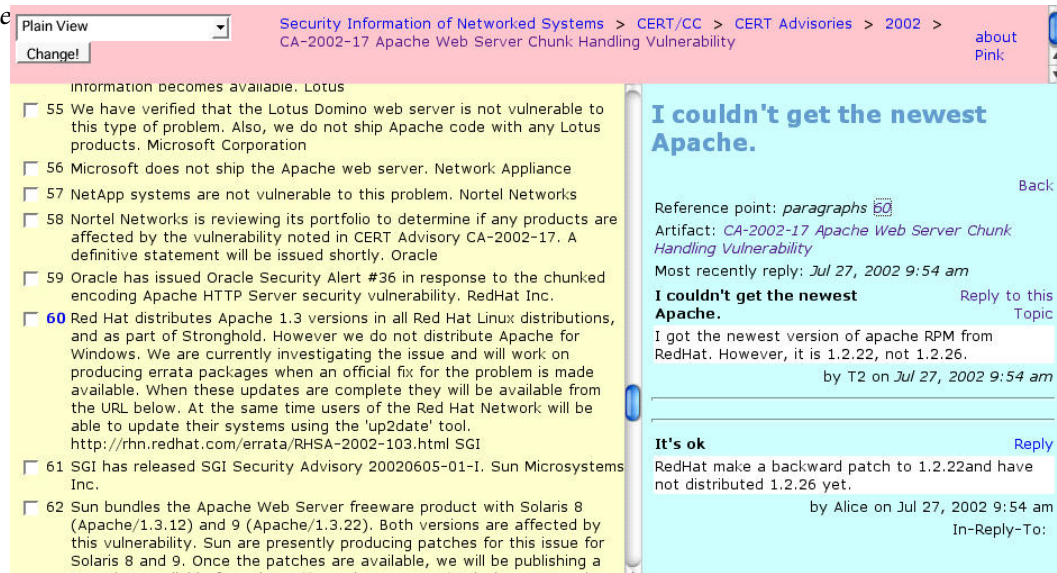


Figure 6. Pink Sample Screen

chitecture that is designed to be extensible and exchangeable. Pink was originally implemented as CGI scripts in the Ruby object oriented language and is currently implemented using the Zope application server. The present implementation is currently under experimental use in our laboratory (including use in preparation of this paper), and has been undergoing testing within an open-source community since March 2002.

The significance of this work and similar work by others goes beyond its potential for improving artifact-centered discourse in online communities. It also represents a better approach to the design of web-based tools for collaboration, in which the design is driven by an understanding of the interactions to be supported.

5. Acknowledgements

The first author was supported by the W. R. Lambuth scholarship of Kwansei Gakuin University. The second author was supported by National Science Foundation grants 9873516 and 0093505. Opinions expressed are those of the authors and do not necessarily reflect policy of the sponsoring organizations.

References

- [1] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9), 1987.
- [2] J. R. Davis and D. P. Huttenlocher. Shared annotation for cooperative learning. In *Third International World-Wide Web Conference*, 1995.
- [3] D. Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems*, 8(1):80–112, 1985.
- [4] M. Guzdial. Information ecology of collaborations in educational settings: Influence of tool. In *In Proceedings of the 2nd International Conference on Computer Supported Collaborative Learning (CSCL 97)*, pages 83–90, 1997.
- [5] B. Leuf and WardCunningham. *The Wiki Way*. Addison-Wesley Longman, March 2001.
- [6] M. Roscheisen, C. Mogensen, and T. Winograd. Beyond browsing: Shared comments, soaps, trails, and on-line communities. In *Third International World-Wide Web Conference in Darmstadt, Germany*, April 1995.
- [7] Sharepoint portal server. <http://www.microsoft.com/sharepoint/>.
- [8] T. Sumner and S. B. Shum. From documents to discourse: Shifting conceptions of scholarly publishing. In *Proc. CHI 98: Human Factors in Computing Systems*, April 1998.
- [9] D. Suthers. Collaborative representations: Supporting face-to-face and online knowledge-building discourse. In *Proceedings of the 34th Hawai'i International Conference on the System Sciences (HICSS-34)*, January 2001. <http://lilt.ics.hawaii.edu/lilt/papers/2001/Suthers-HICSS-2001.pdf>.
- [10] D. Suthers and J. Xu. Kukakuka: An online environment for artifact-centered discourse. In *Education Track of the Eleventh World Wide Web Conference (WWW 2002)*, pages 472–480, 2002. <http://www.c2.com/cgi/wiki?WikiWikiWeb>.
- [11] Tecfa newsletter. <http://tecfa.unrge.ch/>.
- [12] The annotation engine. <http://cyber.law.harvard.edu/projects/annotate.html>.
- [13] Document object model (dom) level 1 specification, 1998. The World Wide Web Consortium <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>.
- [14] K.-P. Yee. Critlink: Better hyperlinks for the www. In *Hypertext 98*, 1998. <http://www.crit.org/ping/ht98.html>.
- [15] Zope (z object publishing environment) community. <http://www.zope.org/>.