CHAPTER 5.3

Annotator: Annotation Technology for the WWW

Ilia Ovsiannikov and Michael Arbib
University of Southern California Brain Project
University of Southern California, Los Angeles, California

Abstract

Annotation technology is a theoretical analysis of annotations. Grounded in studies of paper-based, handwritten notes, it applies the findings to electronic media to produce a set of practical recommendations for annotation software development. In the theoretical section, behavioral, ergonomical, functional, computational, and other aspects are discussed. The implementation part presents Annotator, a system for online annotations on arbitrary hypertext documents which is based on the concepts of annotation technology.

5.3.1 Introduction

There is no doubt that you have read books and taken notes. The custom of annotating is very common, especially during studying, reviewing, proofing, or research. The reader can spend a lot of time and effort on it, and the next time one takes those documents in one's hands, the annotations will readily help refresh one's memory, locate a paragraph of interest, provide proofing instructions, or do one of the many other important things that make annotations indispensable in everyday life.

Consider the following paradigm as a concrete example. In the academic environment, researchers and students annotate papers as a matter of daily routine. Later they write reports, manuscripts for which the annotations are reused. It is the nature of this activity to collect shreds of informations from a large variety of printed publications. Many a time the writer has to go over stacks of annotated papers in search of the particular excerpts of interest. This task can be very demanding and time consuming. This chapter presents the concept of *annotation technology* (AT), which explores various aspects of electronic annotations and offers solutions that can help this process.

Electronic annotations have their roots in handwritten notes; however, advances in personal computing have allowed us to take a step ahead. With the coming of electronic annotations, the student in the above-mentioned paradigm receives a variety of benefits. Annotations and original documents are now just one click away. The student can also easily keep track of the many old papers that would otherwise be half forgotten. Electronic annotations can be searched, shared, and used to construct comprehensive summaries or bibliographies. Applications of electronic annotations are by no means limited to the academic environment. Overall, insights offered by AT can leverage performance and decrease human effort in activities ranging from personal notetaking to office work and publishing.

5.3.2 Overview of Existing Annotation Software

Before we start with the discussion of annotation technology, let us take a brief look at existing annotation software systems. These systems can be roughly divided into two major categories: personal and collaborative. Just as the name suggests, personal systems are intended primarily for individual use. Microsoft Word (Microsoft, 1998) is one example of such software. MS Word allows the adding of typed, handwritten, or voice comments to documents. When the document is opened again in a next session these comments can be examined, edited, or deleted. In this paradigm, annotations become an integral part of the document and cannot be shared among many users.

Annotation sharing is one of the key features that distinguishes collaborative systems from personal ones. Third Voice (1999) represents an example of such a system that has come into existence quite recently. An add-on to Netscape Communicator 4.x, it allows attaching notes to selected parts of hypertext pages. If a note is marked "public," it will also show up on the many screens of other users that come to this page. These users can respond to that note to begin a discussion and thus work together, or collaborate, on some problem.

Multimedia-enabled annotation systems represent a special class of software by themselves. Some of these systems, such as XLibris (Price *et al.*, 1998), focus on the problems of integrating handwritten notes into electronic documents. XLibris accepts pen input that is superimposed on the view of a document. The user can choose between pens of different colors, a marker, and an eraser. Later on, annotation clippings can be reviewed and searched in the Readers Notebook. XLibris also permits creation of various user-defined links that can be traversed to jump to other document parts.

Other multimedia systems such as DIANE (Benz *et al.*, 1997) concentrate on supporting annotations on video, sound recordings, and all other kinds of media. Thanks to a dynamic download of viewer code, DIANE supports annotations on arbitrary types of media. In general, systems of this class often serve one purpose, namely to index the datastream by attaching to it human- or machine-readable labels. This index can be utilized to tag or search data that otherwise are difficult to index.

One more software system that must be mentioned here is EndNotes (Niles, 1998). Strictly speaking, this is not an annotation system per se, but it will be relevant to our upcoming discussion of annotation technology. EndNotes is a personal bibliography database that contains records of various kinds of publications that the user chooses to retain. It is used extensively during composition to find and properly format literature references. For our purposes, EndNote's search represents the feature of interest. We can view the bibliography database as a database containing stand-alone notes along with referencing information. A query into such a database will effectively search annotations. This is not the kind of search available for us when we deal with ordinary handwritten notes, nor is it characteristic of all systems of electronic annotations, but it is central to the design for annotation technology that we present here. For a more thorough review of existing annotation software, please refer to Ovsiannikov *et al.* (1999) which discusses more than a dozen systems, offers a feature-by-feature comparison, and also looks at the implementation side.

5.3.3 Annotation Technology: An Integrative Approach

Annotation technology (AT) is a theoretical analysis of annotations aimed at producing a set of recommendations for the design of electronic annotation software and is the basis for the Annotator system we have developed as part of the USC Brain Project. In this chapter, AT is presented from an integrative point of view. We start by sketching out behavioral aspects of handwritten, paper-based annotations and then make a transition to ergonomic, functional, and computational aspects of electronic ones.

In the beginning, let us introduce some new terminology that will be used throughout the rest of the chapter. Here we should mention that electronic annotations have a concrete foundation in ordinary handwritten, paper-based ones and, perhaps, should be thought of as their extension. Thus we feel it appropriate to accompany this part of the article with examples from paper documents (see Fig. 1). This figure shows the words "frame of reference" selected by means of circling. This is an example of a text *atom*, which is a contiguous portion of text. In general, documents often contain information expressed in forms other than text. For instance, it can be figures, tables, or even, in the case of electronic documents, sound or video. Hence, in order to be more general, we can extend the definition of atom to cover all other forms of presentation as follows: an *atom* is any selected piece of information, which is contiguous with respect to its form of presentation.

For the purpose of annotation atoms sometimes are found aggregated in a collection, as shown in Figs. 2 and 3. Here one can see that the equations containing the first and the second derivative are both selected by means of

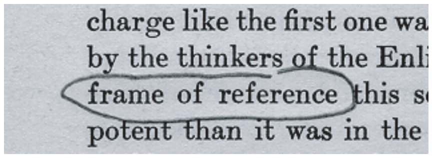


Figure 1 Atom created by circling several words.

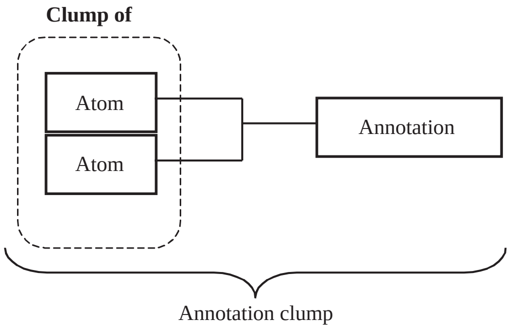
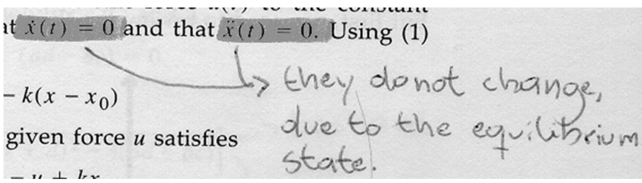


Figure 2 An annotation attached to a clump of atoms forms an annotation clump.


it $\dot{x}(t) = 0$ and that $\ddot{x}(t) = 0$. Using (1)

$- k(x - x_0)$

given force u satisfies

Figure 3 Annotated clump of atoms that were created by highlighting.

highlighting. This allows us to speak of them as atoms. Next, we can say that the two atoms are interrelated in at least two ways: schematically, by the hand-drawn connecting link, and semantically, by the reader's comment referring to both of them. This example illustrates the important concept of a *clump*, which is a set of semantically related atoms linked to the same comment.

Often, the reader not only marks up document portions to create a clump but also attaches to the clump some *annotation*. Let us define an *annotation* to be a datum added by a third party to the original document. From the presentational aspect, this datum can be a handwritten note, a symbol, a drawing, or even a multimedia clip. Fig. 3 shows an example of a handwritten annotation. Notice that the annotation text "they do not change..." connects to its clump by means of an arrow. This arrow represents an example of a *link*. A *link* is something that connects clumps of atoms and annotations together into an *annotation clump* or just *clump*. In Fig. 3, the annotated clump consists of two highlighted equalities and the handwritten text.

Handwritten Annotations

To get a good grip on the new terminology, let us consider some more examples of handwritten annotations. These figures were handpicked from a large number of books that went through students' hands. Annotations from such a source tend to be mostly academic and study oriented but do an excellent job when it comes to demonstrating the multiplicity of annotation form and content.

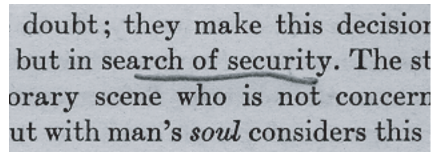
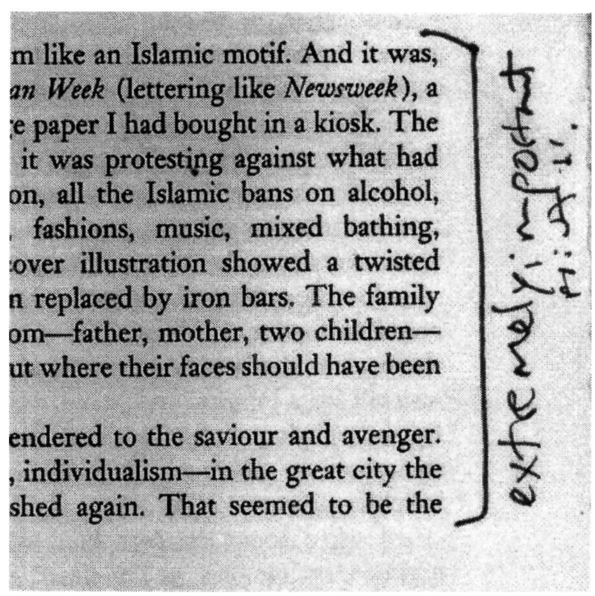

doubt; they make this decision
but in search of security. The st
orary scene who is not concern
ut with man's *soul* considers this

Figure 4 Small atom created by underscoring.

Consider Fig. 1, which shows the words "frame of reference" circled up into a text atom. The reader wanted to select this particular notion out of the document. Because this atom consists of only three words, selecting it by circling was convenient as the method of choice; however, circling is just one out of many methods for selecting short phrases, individual words, or parts thereof. Fig. 3 shows two atoms that are highlighted with a marker, and Fig. 4 presents an example of using an underscore. These selection methods are very popular and convenient as well, but the list is not over. Short atoms can also be enclosed in a box (which is something in between underscoring and circling), struck out, or bracketed or put in parentheses.

What if the text to be selected is longer than just a few words? In such a case, one can continue using the methods outlined above, but the bigger the atom gets, the more handwriting work must be done. When its length is on the order of several lines, *range* selections become the method of choice. Fig. 5 shows an example of a range, which is the bracket located at the right margin of the document. Imagine the atom in Fig. 5 being underscored and you will see that range is a very compact and convenient method. Not only is too much underscoring difficult to carry out, but it also can easily make the document illegible.


m like an Islamic motif. And it was,
an *Week* (lettering like *Newsweek*), a
e paper I had bought in a kiosk. The
it was protesting against what had
on, all the Islamic bans on alcohol,
fashions, music, mixed bathing,
over illustration showed a twisted
n replaced by iron bars. The family
om—father, mother, two children—
ut where their faces should have been

endered to the saviour and avenger.
, individualism—in the great city the
shed again. That seemed to be the

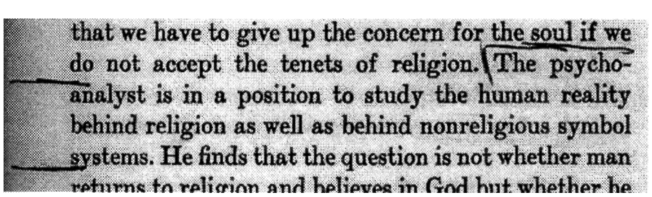
Figure 5 A larger selection uses a bracket to create a closed range atom.

The range in Fig. 5 is called a *closed* range, as it clearly marks its boundaries by means of the angled ends. Other types of text ranges include open, detailed, and hybrid. An *open* range is one that does not set a clear-cut boundary on either side, usually by omitting the angled ends. Such a selection is quite unprecise but occurs commonly. A *detailed* range always marks both of its ends explicitly and precisely. An example of a *hybrid* range is shown in Fig. 6. Here, its starting boundary, "The psychoanalyst," is detailed, as it is specified exactly to within the word. On the other hand, its ending is open, because the markings show that it is located somewhere in the line starting with "systems."

Ranges also differ with respect to the choice of the delineating bracket. Some readers prefer vertical lines or square brackets, while others opt for curly brackets or even tall parentheses. The choice is often a matter of habit, but sometimes it carries additional semantics. Here we will present only one example of markings loaded with semantics, as a longer discussion would be beyond the scope of this chapter. As a rule, there is no second interpretation as to the meaning of text that is crossed out. In editing and proofing, this means that the text atom must be deleted from the document.

Step the atom length up even more and chances are it will not fit a single page. One way to select such big portions of text is to allow ranges to continue onto the next page or pages. Just as in the case with regular ranges, all the range types and markings mentioned above are available here, as well.

Throughout this discussion, we have talked about selecting text by the character or line. These are just two of the four primary *levels of granularity* of selection. On the next level, *page*, the annotator selects atoms by the page. There is no doubt you have worked on this level many a time. One example could be marking pages in the corner or employing a bookmark. The highest level of granularity is the level of the document. We know we are working at this level whenever we affix a post-it note to the first page of a paper or jot a summary in the free space next to the title or even stack up the printed contents of our bookshelf into a "clump of books," so to speak. So far in our atom typology we have concentrated on text atoms only. This type is very widely used and is quite important in everyday work. When it comes to annotating pictures, a different atom type is used. For the reasons of space we will limit our discussion to text only.



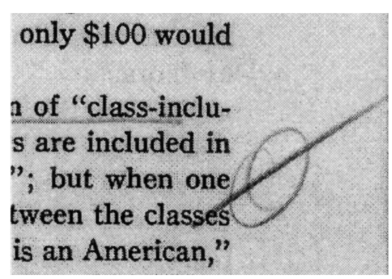
that we have to give up the concern for the soul if we
do not accept the tenets of religion. The psycho-
analyst is in a position to study the human reality
behind religion as well as behind nonreligious symbol
systems. He finds that the question is not whether man
returns to religion and believes in God but whether he

Figure 6 One boundary of every hybrid range is specified precisely while the other is not.

Let us continue on with our discussion of handwritten notes and take a closer look at annotations. Just like atoms, handwritten annotations can be of text or picture type. Fig. 3 shows an example of a text annotation that is attached to a clump of atoms by means of an arrow. By looking at this figure, one can see that the meaning of the attached note is pretty much clear to anyone who is familiar with that document. This is an example of a *common* annotation, as opposed to idiomatic or idiosyncratic annotations. Fig. 7 shows a very different annotation that resembles the "empty set" symbol in form. This is an example of a *pictorial* annotation, which can also be considered *idiosyncratic*, as it looks like it is meaningful only to the person who jotted it down. In between these two poles you can find *idiomatic* annotations, which are characteristic of people of a certain profession (e.g., proofreaders).

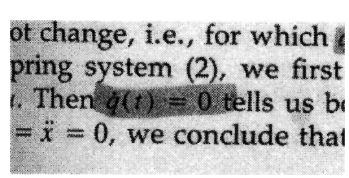
What unites annotations in Figs. 3 and Fig. 5 is that they are both *explicitly* marked. Consider Fig. 8, which shows a single atom created by means of a marker pen and has no note attached to it. Does that render it useless? Not at all. Even when atoms or clumps of atoms have no accompanying notes, the atom markings serve as annotations themselves. In particular, in Fig. 8 the atom's yellow color is equivalent to writing "this is important" next to the selection. Such apparently missing annotations occur very commonly and are called *implicit*. Generally speaking, annotators very often skip putting down notes of particular functions or semantics that occur too often. Such "compression" is performed in order to save time during writing and effort during reading and to avoid clutter.

The implicit annotation in Fig. 8 is conveyed by means of a special color. Are there any other forms of presentation? It turns out that there is a variety of atom markings that convey some definite semantics and,



only $100 would
n of "class-inclu-
s are included in
"; but when one
tween the classes
is an American,"

Figure 7 The meaning of an idiosyncratic annotation is not immediately obvious to anyone but the author.



ot change, i.e., for which
pring system (2), we first
. Then $q(t) = 0$ tells us b
$= \ddot{x} = 0$, we conclude that

Figure 8 Highlighted atom without annotation.

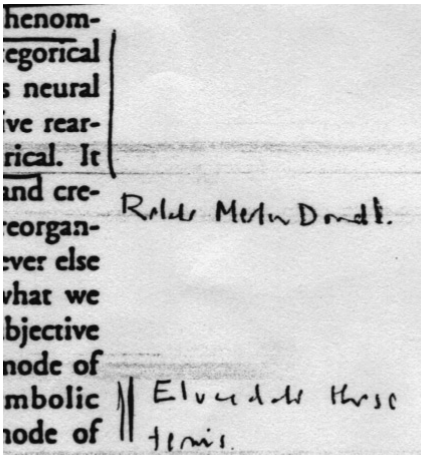


Figure 9 Compared to single, double ranges carry more emphasis.

sometimes, a degree associated with it. Fig. 9 shows two ranges, one of which is *single* and the other *double-line*. In certain cases, the single range may carry "this is important" semantics. The double-line range stresses this statement to make it "this is *very* important." A triple-line range would mean "very, very important" and so on. Sometimes the number of lines is replaced with the thickness of the range or underscore. A thicker selection corresponds to stressing or, literally, underscoring the statement contained in the atom.

Action markings are yet another form of implicit annotations. An indispensable tool of professional proofreaders, it associates actions with various forms of atom selections. For instance, a strike-out atom almost invariably corresponds to "deletion." A vertical line separating two characters often stands for "insert space" and so on.

For a moment let us return to explicit annotations and look at their forms. Fig. 3 shows an example of an explicit *text* annotation, which presents itself as a handwritten note. A different form is shown in Fig. 7. Here the annotation is represented with a *symbol*. Finally, the last but not the least form is pictorial when the reader creates an annotation by means of drawing.

Up to this moment we have discussed explicit annotations and implicit ones which are contained in atom markings. To complete this picture, we must also mention annotations that can be found implicit in links connecting clumps to notes. Suppose that an atom contains some statement, and the reader would like to add a conclusion that follows from it logically. At this point, it can be convenient to use a follows-from mathematical symbol in the form of an arrow to stress the causal relationship between the atom and the note. Thus, the arrow gets to carry semantics additional to its primary linking function.

Just like annotations, handwritten links can be either explicit or implicit. Examples of explicit links are shown in Figs. 3 and 10, where atoms are linked to their notes by means of an arrow. Readers tend to use explicit links

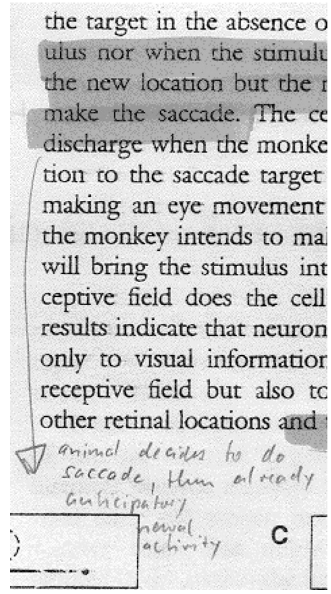


Figure 10 Arrow link connects atom with its annotation.

whenever there is an ambiguity as to which atom a particular note refers to. In Fig. 10, the arrow is necessary to show that the upper atom is the atom of interest and not the lower. Similarly, the arrow in Fig. 3 makes clear that the note refers to both of the formulas.

Links can take forms other than that of a connecting element. In certain cases, annotations refer to atoms by means of a certain symbolic label (e.g., a number). This is the form of choice whenever the annotation must be placed quite far from the atom or clump of atoms and using an arrow is impossible or would create unwanted clutter on the page. Color can also be used as a label to institute a connection; however, in the majority of cases, the link is implicit. The annotation is already connected by being *adjacent* to the clump of atoms. Fig. 5 shows an example of a remark being positioned next and parallel to the closed range. Such a placement tells us unambiguously that the two are directly related.

Distributed Clumps

We have briefly discussed handwritten annotations, their structure, and presentational forms. One important thing must be noted here. As a rule, the examples above deal with atoms and annotations located in the same document and even on the same page. If this restriction is lifted and notes are allowed to be distributed across many printed sources, the annotation clump transcends its locality and becomes *distributed*. Fig. 11 shows an example of this paradigm. In the figure, the note "Experiments from the two papers..." refers to two clumps of atoms in different papers. We can say that, when combined, these two local clumps result in a distributed clump or a set of clumps which share a common annotation.

In general, if we do not limit our scope with handwritten annotations and take into consideration hypertext and multimedia sources, one can hypothesize distributed clumps that cross document boundaries and include as well atoms of type video or sound and so on. As you may have noticed in Fig. 11, notes attached to a distributed clump refer to several papers and at the same time may reside in just one or even none of them. This is unlike handwritten annotations, where notes are almost invariably located right next to the clump of atoms. This is one reason why annotation technology treats annotations and atoms as individual entities. Our Annotation Database is an embodiment of this approach.

Annotation Databases

An Annotation Database (ADB) is a collection of information about the location and contents of atoms, annotations, and their dependencies in the form of links. Fig. 12 shows schematically an ADB divided into two major sections: the clump section and the annotation section. It contains all data about in what document and exactly where in that document a clump is located. ADB also contains links, annotations, and, now optionally, their position in the document.

The ADB effectively takes clumps and annotations out of a document to be stored independently. This approach makes it possible to utilize the information in new ways. One such novel application is *annotation search*. When queried by a user, an ADB will respond with a collection of matching records. Having the information about location of atoms and annotations, the user can easily retrieve the original annotated document and go directly to the clump of interest. To make this paradigm more concrete, imagine ADB implemented as an online hypertext system. Querying such a database would amount to typing keywords of choice in the browser window. Once the request has been submitted and the ADB has returned a list of hits, going to the annotation within an online document would be as simple as clicking a link, as described in the next section about our Annotator system.

Consider the list of hits returned by an ADB. Generated dynamically, this list contains atoms and annotations and can be treated as a new annotation clump. This new clump can be used later on to assist writing a new

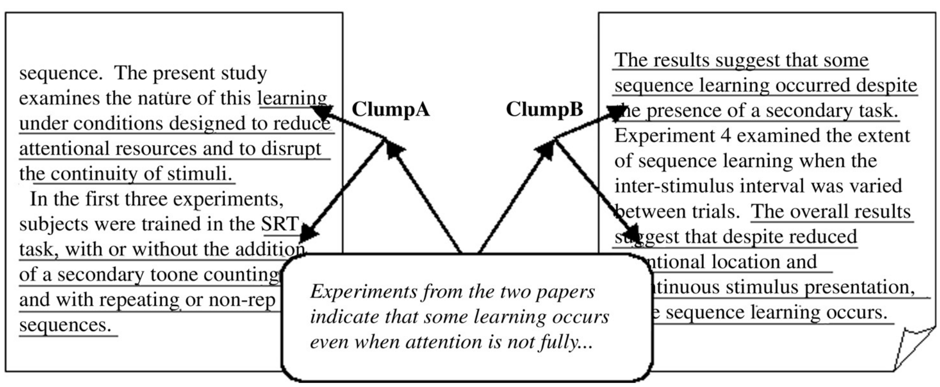


Figure 11 A distributed clump consists of atoms that reside in different documents.

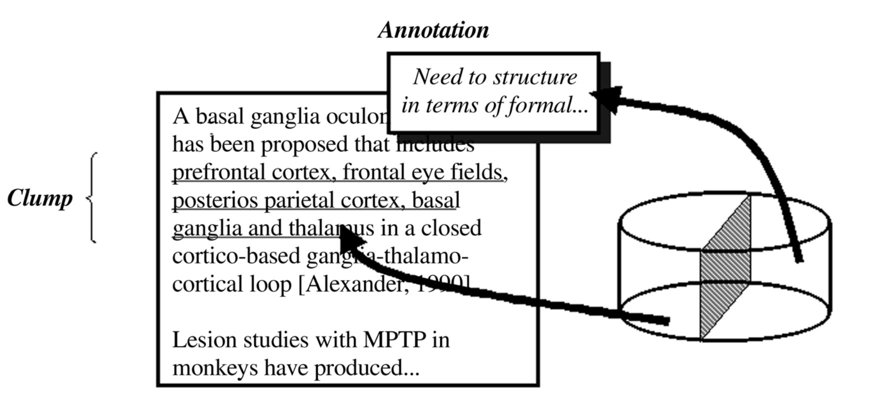


Figure 12 Annotation Database.

manuscript or in putting together a bibliography on a certain topic.

So far we have talked about Annotation Databases under the assumption that they are personal (i.e., individual for each user). What if parts of Annotation Databases were open for access to other people? In this case, ADB will become a shared resource. Access to shared annotations may come in handy when a team of people is working together, such as in an office environment or research laboratory. Opinions, solutions, thoughts, or evaluations—all can be shared concurrently between the members of such a group leading to increased productivity. It should be noted that annotations that are meant to be shared should always be unambiguous; otherwise, ADB users other than the owner may find interpreting the notes a very difficult task.

USCBP Databases: Integration of Independent Sources

Annotation technology research is a part of a greater effort supported by the USC Brain Project (USCBP). This effort is directed at development of a number of on-line databases for the neuroscience community. Consider, for instance, the problem of building databases for online models also known as Brain Models on the Web (BMW). A typical model consists of description, source code, interfaces, executables, and such and resides in the BMW database; however, having just these data for modeling is insufficient. First of all, the input and output data patterns and time series are indispensable elements. A good model should also refer to literature that supports or discusses relevant issues, assumptions made during development, and so on.

The USCBP has been developing databases to address these needs. The Time-Series Database (TSDB) has the capability to store and retrieve various patterns, electrode recordings, and images. On the other hand, the Summary Database (SDB) contains text summaries, notes, and logs. In this framework, AT is positioned to solve the important problem of integrating these databases. This capability for integration of independent information sources stems from the notion of distributed clump. With AT tools, the user has the freedom to impose arbitrary fine-grained structure on documents without actually modifying those documents. The functionality of annotations is also extended to include pointers, as in hypertext (see, for example, Inso, 1998). Taken together, these functions allow the models in BMW to be easily linked to supporting statements in documents and SDB summaries and to datasets in the TSDB.

5.3.4 Annotator

In this section we will introduce Annotator, a pilot implementation of the principles of annotation technology. Annotator is a distributed collaborative software system for making online annotations on the World Wide Web. Written completely in Java, it works with Netscape 4.x on any platform.

User Interface and Functionality

Working with Annotator is as easy as using Netscape Communicator to browse the Internet. By default, Annotator resides in the "disabled" mode. In this mode, all its features are disabled and the browsing is no different than if Annotator were not installed. To enable annotation features, the user must open a special URL. In response, the browser will prompt for a username and password to log in to an Annotation Database. (Some of these details of access to, and use of, Annotator will be described in the "Available Resources" section; here, we focus on the general functionality of Annotator.)

From this point on, all retrieved pages will show up along with annotations you or someone else created earlier. Fig. 13 shows an example of annotations embedded into a page. Atoms are shown in blue, and notes are shown in red. All atoms and annotations will also show up in another window called the Database Control Panel

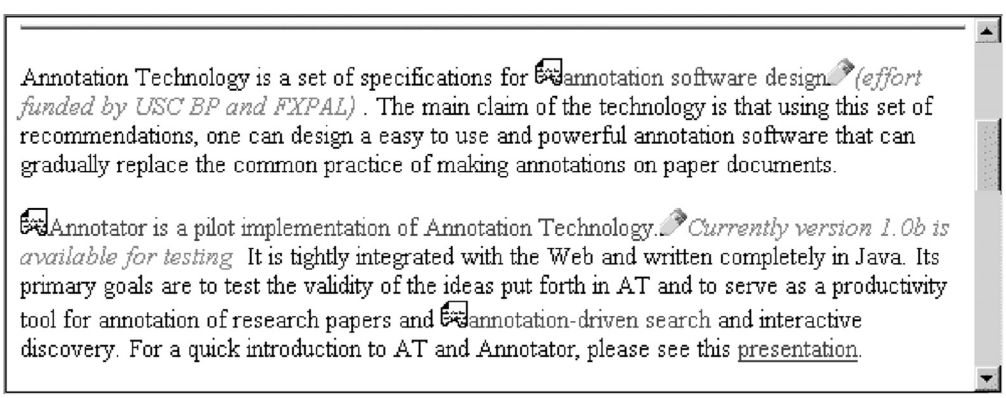


Figure 13 Viewing annotations in a browser window.

(DCP; see Fig. 14). The DCP displays annotations in a tree-like structured form, where branches correspond to links between atoms, clumps, and notes. The user can browse through such a tree by expanding or collapsing certain items. Double-clicking on an item causes the browser to open a page and scroll to the annotation of interest. Certain advanced features such as linking clumps and atoms by dragging and dropping the icons are in development. For a moment, let us return to Fig. 13, which shows a sample annotated page. On that page, the symbolic icons associated with atoms and notes work as miniature buttons. Clicking on one will make the DCP annotation tree change to reveal the chosen item.

The user can annotate a page by invoking the Netscape's HTML editor, Netscape Composer. Fig. 15 shows a screenshot of a page being annotated. From the editor's menu, the user commands a wide variety of annotation-related actions. For instance, to create an annotation, the user selects an atom of interest, clicks "Annotate," and types the note in the space allocated by the editor. A future version of the software will allow adding more atoms to a clump by making a selection and choosing "Add Atom." Long, out-of-line annotations called *memos* can be created in a manner similar to regular annotations. In the end, the user saves the work to the Annotation Database by clicking on the editor's "Publish Page" button.

One feature that is central and in some ways unique to the design of Annotator is the ability to search and summarize annotations. This is where the complexity and power of the Annotation Database comes to its full realization. To query ADB, the user must bring up the Database Control Panel, select the search mode, and type in a set of keywords. In return to this query, ADB will retrieve matching atoms and notes and display them in the tree form. By clicking on these items, the user now has direct access to the annotations in the original online documents. As an alternative, the user can select a number of retrieved items to create a distributed clump. This can be useful, for instance, to collect together annotations related to the topic of interest across various research papers. The user can also copy-and-paste such a clump in a word processor. In this case, Annotator will combine selected annotations and atoms into a readable text summary to be printed out or edited into some text.

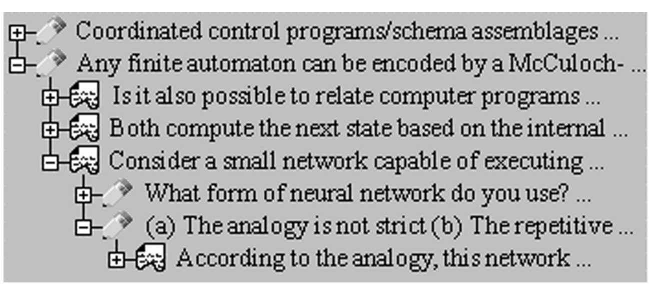


Figure 14 DCP annotation browser.

Implementation

Having discussed Annotator's functionality and user interface, let us take a look at the implementation details. Fig. 16 shows a diagram of the main system components. The proxy and Annotation Database comprise the system's core. Together, they are designed as shareable resources and can run on either a stand-alone server or on the user's PC in the single-user case. The browser's functionality is augmented by a plug-in package that must be installed before use in each workplace.

Consider a typical scenario where the browser issues a request to retrieve a hypertext document. In the disabled mode, the proxy acts as a repeater that forwards the request on to the server and passes back a response containing the page. In this mode, the user can browse the Web as if Annotator were not installed at all. Once the user switches the proxy over to the enabled mode by logging in, the pattern of communication changes. Every time the browser issues a request, the proxy will forward it on to the server and at the same time query the database for any annotations and atoms associated with the given URL. Once the server returns a document, the proxy checks the search results. If the database returns a non-empty set of records, the proxy will dynamically parse the document and insert the annotations and atoms at their proper locations. After this, the document is passed on to the browser for display. Note that the insertion is made in generating this display, not on its source document. This is a distinctive feature, for it

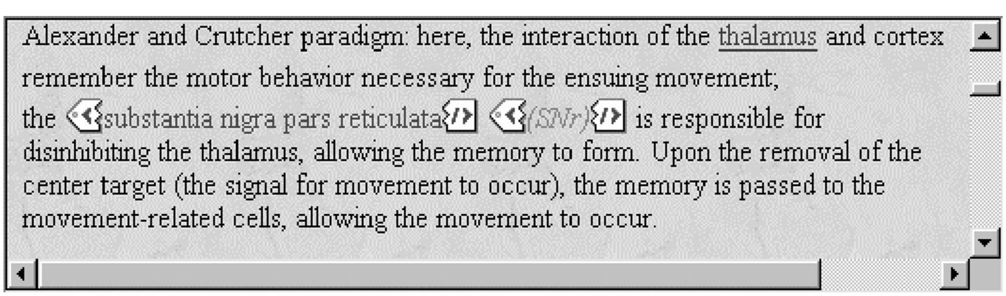


Figure 15 Editing annotations in Netscape Composer.

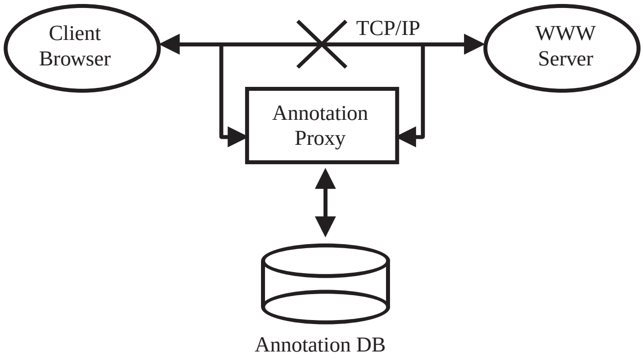


Figure 16 Annotation proxy queries ADB and dynamically inserts annotations into hypertext documents.

allows the user to make or view annotations on documents whose files are locked or otherwise unavailable to change. In general, downloading and storing any portions of document without permissions violates the copyright law. That gives one more reason why clumps must be annotated *in situ* rather than copied to the user's computer.

The Annotation Database is a custom-made Java-based database. To minimize the overhead due to data retrieval, it is designed for maximum real-time performance. The ADB schema contains several tables, three of which are most important. These are the atoms, annotations, and links tables. Along with the text contents and associated URL, the atoms and annotation data structures store information about their location within the documents called *anchors*. According to the schema, every annotation has one anchor assigned to it. Every atom has two anchors, one for the beginning of the selection and one for the ending.

A number of considerations are relevant to the implementation of anchors. There is no guarantee that an annotated page will never change. At any time, the owner of the page can edit it, and some pages are generated dynamically and may differ from one download to another. Hence, it is highly desirable that the system is robust with respect to small changes in hypertext pages and that the anchors in Annotator are implemented with this consideration in mind. To record a location in a page, Annotator finds a reasonably short but unique substring at that location. After that, the substring undergoes hashing and the hashcode is stored in the record along with the string's length. Anchors also store the first few characters of the search string. This is done in order to make finding the location during retrieval computationally more efficient.

The ADB is designed to be a shareable component. Because HTTP is a stateless protocol, the proxy must simulate a database session for each user from the moment users log in and up to the point when they log out. To this purpose the proxy maintains a table of sessions listing all database connections that are currently active. If any connection has not been active for a certain amount of time, the proxy will close it, effectively logging out that user after a period of inactivity.

The Annotation Database supports annotation searches. It takes a set of keywords as input and returns a collection of matching records. The search can be restricted to atoms or annotations only. Annotations are added by means of submitting the annotated document to the proxy. When the user clicks the "Publish" button in Netscape Composer, the editor uploads the page with annotations to the proxy. In its turn, the proxy intercepts the page, parses out any annotations and atoms contained there, computes anchor information, and saves all the data in the database along with the page's URL.

In this chapter, we have presented annotation technology, an integrative theory of annotations that offers a systematized set of recommendations for the design of advanced electronic annotation software. Based on an analysis of handwritten comments, AT proposes the annotation-clump framework that conveniently represents annotations as data structures. Annotations acquire new functionality during the transition from handwritten to electronic media. In relation to this fundamental process, AT introduces the notions of distributed clump, Annotation Database, and annotation search.

Available Resources

The paradigm described above has been put to the test with the development of the Annotator software system that is available for download from the USC Brain Project server at http://www-hbp.usc.edu/Projects/annotati.htm. The software consists of two packages, client and server. The client package must be downloaded and copied into the Netscape 4.x plug-in directory as explained in the accompanying manual. After downloading and unzipping, the server can run on any computer with Java Virtual Machine and network access. Note that the server can be used individually as well as shared. To log in to the Annotation Database, open URL http://annotation-proxy/login and log in as *admin* with password *admin*. To log out, go to http://annotation-proxy/logout.

Acknowledgments

NIMH grant MH 52194–03 and a grant from Fuji Xerox Palo Alto Laboratories supports this research. We thank Thomas McNeill, for support for the early stages of this work, and Abhitesh Kastuar and Shivani Aggarwal, for their work on programming Annotator.

References

Benz, H., Fischer, S., Mecklenburg, R. *et al.* (1997). DIANE: hypermedia documents in a distributed annotation environment. In *Proceedings of HIM '97*. Dortmund, Germany.

Inso (1998). *DynaText Professional Publishing System*, http://www.inso.com/dynatext.

Microsoft (1998). *Microsoft Word.* http://www.microsoft.com/office/word/default. htm.

Niles (1998). *EndNotes 3.0*. http://www.niles.com/.

Ovsiannikov, I., Arbib, M. A., and McNeill, T. H. (1999). Annotation technology. *Int. J. Human-Computer Studies*, 50.

Price, M., Schilit, B. N., and Golovchinsky, G. (1998). *XLibris: The Active Reading Machine*. ACM Press, Los Angeles, CA.

Third Voice (1999). *Third Voice*. http://www.thirdvoice.com.