Draft version

A visual approach to versioning for text co-authoring

José A. Pino

A way to present alternative pieces of text in multi-author documents is discussed. It is called 'Stick-On' and allows an unlimited number of versions of paragraphs, sentences, words or characters to be considered in the context of the rest of the article. The device is well-suited for collaborative co-authoring. A distributed system incorporating this tool is presented.

Keywords: human-computer interaction, computer supported co-operative work

A basic objective of computer supported co-operative work (CSCW) systems is to provide tools to allow a group of people to accomplish a joint task and produce high quality output (Ellis *et al.*, 1991). This is no easy undertaking, considering the co-ordination required to let users benefit from mutual interaction and avoid undesirable social and system-induced side-effects.

If we focus our attention on supporting a group of people that write a text article together, there are still many design problems to be solved: how are people to co-ordinate themselves for this activity? Are we going to provide tools to negotiate? How is the decision process to be supported?

One way to start is to analyze the time/space dimensions in which the collaboration will take place. Users may work at the same time (synchronous) or at different times (asynchronous). On the other hand, people can be collaborating at the same location (face-to-face meeting) or may be situated at different sites (distributed meeting). A co-operative process may be any of the four combinations.

Another dimension for analysis is to examine the roles assigned to the various group participants. The group can be organized with clear roles, such as the ones defined by Fish *et al.* (1988): co-author, commenter and reader. A second example of roles is to have a single person responsible for the actual writing (scribe) and the rest of the group contributing with ideas, comments and suggestions (consultants). A third option is to have a single role of co-author taken by all

Depto. de Ciencias de la Computación, Universidad de Chile, Casilla 2777, Santiago, Chile. E-mail: jpino@dcc.uchile.cl

participants, as is the case for the production of many scientific articles: each co-author is responsible for contributing text, commenting on the other co-authors' work and improving the writing style.

Neuwirth *et al.* (1990) have reported that people, regardless of their role, want the ability to rewrite pieces of text and not simply annotate them. It is argued that in many cases, when people are reviewing a piece of text, they often consider it more efficient to rewrite the text instead of trying to diagnose the deficiencies of the original.

Efficient methods that facilitate co-operative authoring may also reduce conflict within the group. Take, for instance, the case of an author who is annoyed because a co-author has replaced his carefully worded introduction with one that he or she considers inferior. Even mere comments can hurt the feelings of such an author. The remarks do not need to be derisive or sarcastic to hurt the original author's feelings. So many problems could be avoided if the second (reviewing) author had a simple mechanism for writing his or her own version of the text.

Versions can also be helpful to promote constructive conflict to prevent the poor solutions generated by 'groupthink' as mentioned by Sharples *et al.* (1993). In fact, they suggest that a CSCW system for collaborative writing could be designed to elicit constructive criticism through "mechanisms for enhancing anonymity in idea-generation phases, mechanisms to allow alternative documents to be developed (and thus for conflicting viewpoints to be expressed), or through notations to present arguments in a pictorial or graphical form".

An interesting experience with children trying to carry out joint text authoring with a collaborative tool is reported by Mitchell *et al.* (1995). The tool had essentially no versioning features: the shared view of a document was implemented with replicated copies in each machine and this provoked continuous confusion to the users.

This paper does not attempt to have answers to all questions concerning the collaborative authoring of an article. Rather, it is intended to present a proposal to handle local versions of text pieces using a device called a 'Stick-On'. The ideas have been implemented on a collaborative text writing system called SHADOW which runs on a network of Sun workstations.

Previous work

Joint writing of a piece of text involves a great deal of thinking by the various individuals involved. However, thinking is not a sequential process. As Conklin (1987) has put it: "Thinking seems to proceed on several fronts at once, developing and rejecting ideas at different levels and on different points in parallel, each idea depending on and contributing to the others." There appears then the need to keep those ideas or preliminary writings in any system intended to support the creative part of joint writing.

One way to make visible the modifications incorporated by one person on a draft produced by another one is to support 'change bars' to indicate the points where there is something new (Irish and Trigg, 1989). Using change bars, a co-author can add the new version of a paragraph or sentence below the original one. This has the advantage of displaying all versions simultaneously (if they fit inside

the work window) although it fails to easily show exactly which words or phrases have been changed. It also becomes cumbersome with relatively large pieces of text being versioned.

Another contribution to this field is the PREP editor (Neuwirth *et al.*, 1990). Here, the users are provided with a grid of text chunks, each column being used for a certain purpose; for instance, three columns may be labelled 'outline, content and comments'. Of course, one or more columns could be assigned to hold different versions of the contents. Now, horizontal space limits may place restrictions on the number of versions to be simultaneously visualized, but it is a useful device if there are only a few versions.

Neuwirth *et al.* (1992) introduce the idea of flexible 'diffs': differences between versions are shown to the reader with visual marks. For instance, insertions may be shown in italics, and deletions as underlined text. Users may control which changes are shown and when they are reported. In the case of several versions, it may be difficult to visualize all of them.

Minor and Magnusson (1993) propose a system based on active diffs: a piece of text replacing a previous one is shown underlined, whereas a deletion is depicted as a minus sign and an addition as a plus sign. The plus and minus signs can be expanded to show the actual additions and deletions. An evolution graph helps to visualize the version's history.

Hypertext can certainly be used to hold versions. It is implicitly mentioned in the previously mentioned article by Conklin (1987) in discussing an example consisting of a Design Journal. For instance, a new version of a phrase can be painted to be delimited which would signal that there is another version of it. Of course, additional versions can be stored in successive hypertext nodes. The sequence of versions can be easily followed.

Delisle and Schwartz (1987) introduce the concept of 'context' to hold a version of consistent hypertext nodes and links. Each context can be developed by a co-author. Afterwards, these contexts can be merged with a master context.

Stick-Ons for local versioning

A version of a text piece should be stored in the right place with respect to the rest of the text. Thus, it could be read in the appropriate environment. Of course, the tentatively replaced text should be kept and some visual clues must be incorporated in order to delimit the current version and to signal that this is an optional replacement.

These requirements can be nicely satisfied incorporating the metaphor of the Stick-On: a virtual piece of paper virtually glued over the text to be versioned and intended to contain the new version. Inspiration for the idea came from the Post-It[1] notes. However, instead of being rectangular cards of a fixed size whose objective is to hold temporal comments, Stick-Ons adjust themselves to the size and shape of the new text version. The new version does not have any restriction related to the length of the overwritten text: in the SHADOW system, the text of

[1] Post-It is a trademark of 3M Corporation.

the new version and the text following it are scrolled to make them visually continuous. Contour and colour help to make Stick-Ons easily distinguishable.

To create a new version, a user simply has to select the text to be covered by a Stick-On and then choose a 'new' option from a menu. Automatically, a coloured small square appears instead of the selected piece of text. The user can then type the new text, and the Stick-On size grows to contain it.

Of course, Stick-Ons can be placed on top of other Stick-Ons. The number of Stick-Ons that can be placed over a piece of text is unlimited, thus providing multiple versions to co-exist. Furthermore, the overlap can be partial; therefore, any text piece, even a single character, can be part of co-existing versions of words, phrases, paragraphs. Moreover, users are unaware of the 'version' concept and do not have to be concerned with it. They naturally place coloured 'patches' over text that they wish to modify.

Visually, the whole text can be read with the last versioned piece of text correctly inserted in place. Reviewing previous versions is achieved by simply clicking the left-most mouse button while the cursor is on any place inside the Stick-On; the system then shows the immediately previous version shading the area covered by the removed Stick-On. This may be considered a metaphor of the 'glue debris' left by Stick-On removal.

Stick-Ons can also be placed over null text, allowing the insertion of new words or phrases in the middle of a paragraph. This feature helps users to take notice of changes from last time they read this text section. This is useful even in a single-user situation, but it is very important in a distributed collaborative session.

In a co-operative environment, users do not have to worry about erasing others' contributions. They still have to agree on the best version to keep, but that does not have to be done in the same session. In the SHADOW system, when users have decided on the best version of a paragraph, one of the users manipulates Stick-Ons so that the preferred writing of the paragraph is visible on her screen. She then chooses the "Clean" option from a menu, and that has the effect of removing all Stick-Ons and shades. Using a metaphor, it is like photocopying the paragraph.

Colour may be used to distinguish one Stick-On from another. In SHADOW, there is a provided (but user-modifiable) scale of colours, so that the first Stick-On within a certain paragraph gets a pre-assigned colour, the second another one, etc. This is useful when Stick-Ons are contiguous, and also helps to give a visual hint to the history of versions.

Collaboration using Stick-Ons

SHADOW allows both synchronous and asynchronous collaboration in a distributed setting. Automatic paragraph locking and unlocking are provided to prevent undesirable effects from simultaneous editing.

The following example illustrates the use of Stick-Ons in a real co-authoring process. A group of peers want to co-author a scientific paper. Perhaps they have a face-to-face meeting and they write an outline of the paper as a result. One of the authors expands one or more chapters, the manuscript goes to another author, etc., in a sequential manner. Using SHADOW, each of the authors may cover a single line of the outline with a Stick-On containing a complete chapter. An author

who wishes to propose an alternative writing of a sentence or paragraph does so on a Stick-On which does not delete the previous contribution. Figure 1 shows the state of part of a manuscript before an author attempts to improve it (manuscript example taken from the co-authoring of Baeza-Yates *et al.*, 1995). The author then makes changes using Stick-Ons, and the result is shown in Figure 2. Suppose the proposed changes were not the best ones. Fortunately, the original text can be easily and selectively retrieved 'ungluing' the Stick-Ons (Figure 3). At any time, synchronous work can occur, for instance, if the deadline to submit the paper is approaching. One or more face-to-face brief meetings could be useful to decide on the best writing style and details with the versions-rich manuscript at hand.

A second example of the uses of Stick-Ons is in a variation of the previous case: suppose that a paper needs to be presented to different audiences. For instance, the paper is to be published in a scientific journal, but a variant of it is to be prepared for submission to a trade journal. Of course, the second version can be done without Stick-Ons by simply making a copy of the original article and editing it directly, but the changes will not be apparent. Using SHADOW, the Stick-Ons will clearly mark the places where changes in the writing style or contents have been introduced.

A reviewer's role can also be supported with Stick-Ons. Consider the case of an English teacher correcting a student's précis submitted by electronic mail. Spelling and grammar mistakes can be covered with Stick-Ons showing the correct writing in context. The student can uncover Stick-Ons to see what was wrong. The reviewer can also find the means to state comments. In SHADOW, the comments are placed in 'notes', which are small windows normally closed. A closed note is visualized in the text as a pin. Clicking with the mouse on a pin head opens the corresponding note window. Thus, these remarks are unobtrusive to normal reading, but they may be easily accessed if desired. Notes can be placed anywhere, as can Stick-Ons.

§
The Chilean economy has been steadily improving during the last ten years (see Table 1), growing roughly 80% in the decade. In some publications, it has been mentioned as the sixth Pacific tiger. Even though this is an impressive record, Chile is still a developing country with 4 million people (30% of the population) considered "poor" according to official figures.
§
The country had, until 1973, a long democratic tradition. However, this was lost due to a military coup led by the right wing general Augusto Pinochet, who ruled until 1990. At that time, he passed the power to president Patricio Aylwin, a cristian democrat, belonging to the largest political party in the country. General Pinochet left the country with a booming economy, based on exports.
§
During his four year government, president Aylwin put special emphasis not only on the economic stability but also on the political stability of the country.
§
In December 1993, Eduardo Frei, also from the cristian democrat party, was elected, and in March of this year, president Frei took power for the next six years. Until now, he has followed the same policies left by his predecessor. Therefore, it is expected that the country will have an economic and political stability during the rest of this decade.
§

Figure 1. Initial manuscript

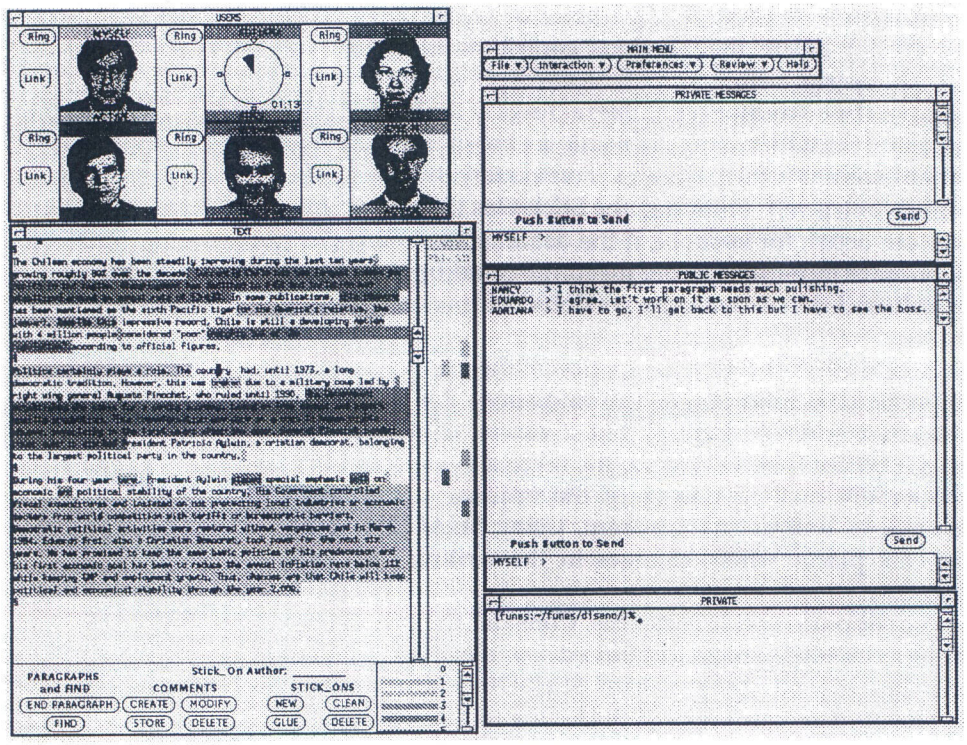

Figure 2. SHADOW user interface showing manuscript with Stick-Ons

Designing usability with the Stick-On metaphor

In some collaborative environments, it may be convenient not to display the author of each contribution in order to facilitate consensus forming, since people would not insist on their 'own' idea. In other cases, it is necessary to keep the identity of each contributing writer as a way to motivate participation. In the latter case, for instance, the SASSE editor (Posner and Baecker, 1993) uses colour for this purpose.

§
The Chilean economy has been steadily improving during the last ten years (see Table 1), growing roughly 80% in the decade. In some publications, it has been mentioned as the sixth Pacific tiger. Even though this is an impressive record, Chile is still a developing country with 4 million people (30% of the population) considered "poor" according to official figures.
§
The country had, until 1973, a long democratic tradition. However, this was lost due to a military coup led by the right wing general Augusto Pinochet, who ruled until 1990. At that time, he passed the power to president Patricio Aylwin, a cristian democrat, belonging to the largest political party in the country. General Pinochet left the country with a booming economy, based on exports.
§

Figure 3. Stick-Ons removed from an edited manuscript

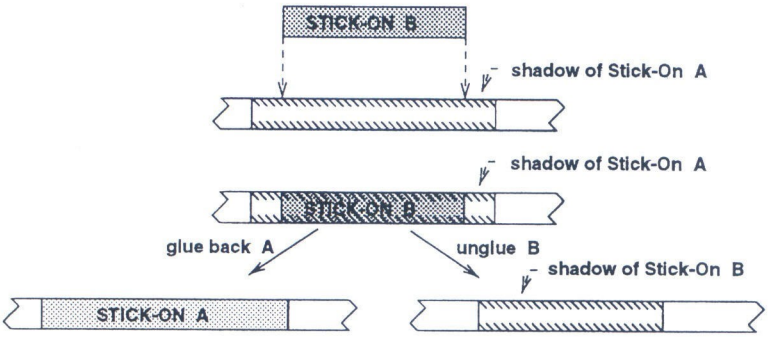

Figure 7. Gluing Stick-On B completely inside shadow of A

Comparison with related work

There are many previous works in which the spatial arrangement of objects on a computer screen indicates relationships among the objects. Borges and Pino (1994) describe some of these relationships when the objects' shape is that of a card. Virtual Post-It notes have been included in other systems with their original meaning of holding comments and with the traditional shape (e.g., Shipman *et al.*, 1995).

It is useful to compare Stick-Ons with other approaches for local text versioning. Stick-Ons do not have the problem of windows running out of visual space, as do solutions that place versions next to one another. Stick-Ons have also an advantage over diffs: they ease visualization when there are several versions. However, the 'versions evolution graph' proposed by Minor and Magnusson (1993) is a more explicit history device than the colours of Stick-Ons. Furthermore, there is no natural ordering of hues, so users cannot immediately know the history of the Stick-Ons without looking at the hues sequence scale at the bottom right of the window. (Repeated use of the system will familiarize the user with this scale.)

Hypertext solutions to versioning are comparable to Stick-Ons when there are various versions for exactly the same piece of text. For instance, when several alternatives have been proposed, in hypertext the user can easily navigate the various nodes containing the versions, with roughly the same effort as ungluing Stick-Ons, except that Stick-Ons will show each alternative version in 'most recent first' sequence in the context of the remaining text. However, when the versioning overlaps or corresponds to portions of an already versioned piece of text, Stick-Ons are clearly advantageous.

Use of Stick-Ons in practice

Stick-Ons are the versioning device provided by the SHADOW experimental joint authoring system. The SHADOW system does not include tools appropriate for group members to decide on preferred versions. The corresponding negotiation and decision are left for a face-to-face meeting to be held after the options have been proposed. SHADOW is also unaware of formatting instructions.

The photocopy tool can be considered not only as a way to have a clean, final

piece of writing to be exported to a formatter, but also as a useful device during intermediate stages. If the users believe they have a consistent non-final paragraph or document but full of now unnecessary Stick-Ons, they can photocopy the text and save it for future reference. Afterwards, Stick-Ons can be placed on the photocopy itself. However, the current SHADOW implementation does not provide a way to keep track of these complete non-final documents. It must be mentioned that the 'global' versioning technique is often used in the field of software engineering to manage source code versions, e.g. in SCCS (Rochkind, 1975) and RCS (Tichy, 1985). A new SHADOW release incorporating features for global versioning is being planned.

The solutions provided by the current SHADOW for author's identification (if needed) are not completely satisfactory. This is clear when other Stick-On attributes, such as time of creation of the Stick-On, are called for. A more general solution is to have an *about* button at the lower end of the window so that information concerning a current Stick-On can be displayed if the button is chosen. Again, this feature is being planned for the next release of SHADOW.

Comprehensive future group experimentation with the system should validate our hypothesis that author identification is not the most important Stick-On attribute (several researchers have argued, in fact, for anonymous contributions). In single-user situations, of course, identification of the author is useless.

A practical worry is also the number of colours to be assigned to Stick-Ons. As mentioned, the versioning system assigns different colours to each successive Stick-On within each paragraph. Theoretically, this ensures that if two Stick-Ons are contiguous or overlapping, their colours will be different (Thorell and Smith, 1990, suggest that if the purpose is to differentiate information, any number of colours is acceptable as long as there are sufficient contrast differences). In practice, after a number of colours, individual users may start to have difficulties discriminating new colours from the previous ones. In SHADOW, after a user-defined number of Stick-Ons have been created for a paragraph, a message is displayed suggesting that the paragraph be photocopied, if possible. The present SHADOW has a palette of 20 colours, in which hue and saturation have been assigned to distinguish one colour from another and to ease text readability when using them for background under the black letters.

Conclusions

The main feature of the Stick-On proposal is its intuitiveness: users are not required to learn concepts such as versions, or manipulate complex tools. The idea of gluing and ungluing colour patches handled with mouse clicks is simple enough to allow immediate use by people accustomed to page editors.

It is interesting to note that the use of Stick-Ons does not prescribe clearly marked stages of divergent/convergent collaborative work, as contexts do. Co-authors may add a contribution at any time, provided a final decision has not been made concerning the article.

We have performed limited experimentation with regular computer users making synchronous use of SHADOW. There was no previous training in this system at all. The basic Stick-On functionality and the photocopying metaphor

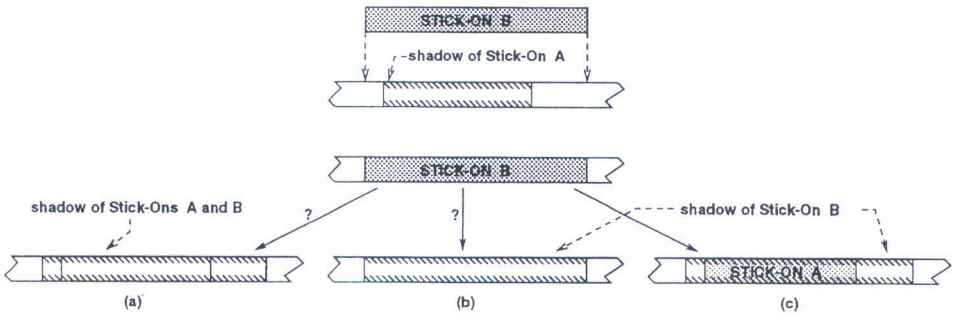

Figure 4. Removal of Stick-On B, placed over shadow of A

As Stick-Ons are coloured to display local versioning history, colour cannot be used in SHADOW to keep track of authors' identities for each version. Two optional solutions to this problem have been designed. The first is to allow users to identify themselves in a 'note' attached to each Stick-On. This is a specialized use of notes but it has two disadvantages. First, it requires that Stick-On users explicitly state their names for each contributed Stick-On. Second, it is a deviation of the original purpose of notes as holders of user comments.

The second solution provided by SHADOW is to display the author's name at the bottom of the window when a Stick-On is made 'current', i.e., when the cursor is over the Stick-On. This feature can be disabled according to user preferences. Further discussion of this issue is presented in a later section.

Another design problem posed by Stick-Ons is caused by overlaps. Before discussing this problem, we need to make some assumptions. It is an obvious requirement for Stick-Ons that reviewing them should not cause information loss. Also, a decision has to be taken about the meaning of a shadow with a vertical line boundary in the middle: the most intuitive meaning is that two contiguous Stick-Ons have been removed for examination of whatever is underneath and the line marks the separation of both Stick-On shadows. We adopt this interpretation.

We may notice now that, if some care is not taken with the implementation of Stick-Ons, there could be information loss or weird semantics. Let us consider first a minor problem illustrated in Figure 4. A user has removed Stick-On A and has decided to glue Stick-On B completely covering the shadow of A. Suppose the system allows this (middle picture in Figure 4). Subsequent removal of Stick-On B poses a problem. If the system depicts the shadows of both A and B, the user will be misled and interpret the shadows as belonging to contiguous disjoint Stick-Ons (Figure 4(a)). On the other hand, if the system shows only the shadow of B, the user will never be able to see the contents of Stick-On A again (Figure 4b)). A simple solution is to make the system automatically glue Stick-On A back before placing Stick-On B on the text. If the user now removes B, he or she will see Stick-On A and the shadow of B (Figure 4(c)).

Let us consider another case (top part of Figure 5): Stick-On B partially overlaps the shadow of A. If this is allowed and the shadow of A is eliminated (middle part of Figure 5), the user will never be able to glue Stick-On A back again, since removal of B will only depict the shadow of B. Showing the shadows of both A

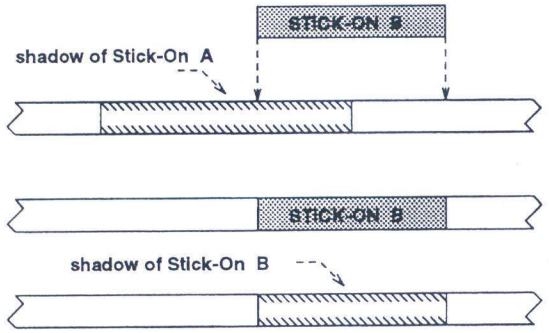

Figure 5. False solution to partial overlap of Stick-On B and shadow A

and B is not possible in this case, as contiguous shadows would appear, and these would have a different meaning.

A solution to this case is simple, but the user must be made aware of it: the addition of a Stick-On with partial overlaps is forbidden. In order to build a version in this case, then, the user has to add a bigger Stick-On in such a way that it completely covers the shadow of the previous Stick-On. This reduces this case to the one discussed before (Figure 4).

Another solution to the partial overlap problem is to make the system automatically enlarge the Stick-On for the user. This solution is shown in Figure 6. The system copies the contents of the shadow to be covered by the new Stick-On. The user, with this solution, is allowed to use Stick-Ons in any situation, but she must be aware that in some cases, the system will enlarge the Stick-On that he or she is attaching.

The last case of overlap — a new Stick-On completely included in the shadow of a previous Stick-On — does not pose any difficulty (Figure 7). The only care needed is to distinguish graphically a Stick-On over a shadow (middle of Figure 7) from a shadow over a Stick-On (Figure 4(c)). This can be done by displaying different textures for the shadows.

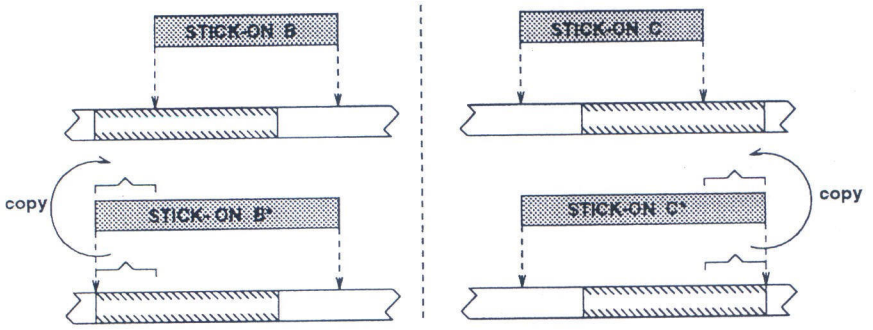

Figure 6. Automatic enlargement of Stick-Ons

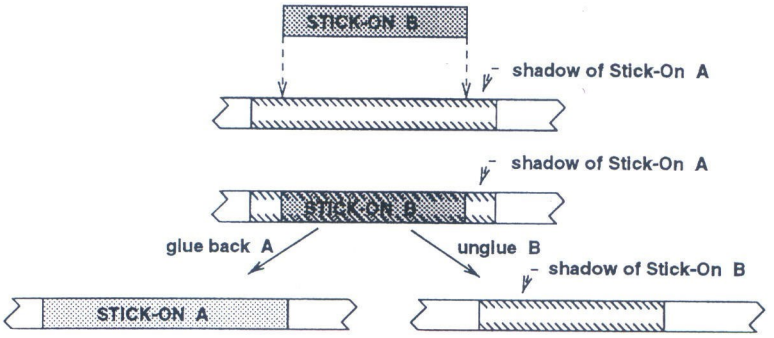

Figure 7. Gluing Stick-On B completely inside shadow of A

Comparison with related work

There are many previous works in which the spatial arrangement of objects on a computer screen indicates relationships among the objects. Borges and Pino (1994) describe some of these relationships when the objects' shape is that of a card. Virtual Post-It notes have been included in other systems with their original meaning of holding comments and with the traditional shape (e.g., Shipman *et al.*, 1995).

It is useful to compare Stick-Ons with other approaches for local text versioning. Stick-Ons do not have the problem of windows running out of visual space, as do solutions that place versions next to one another. Stick-Ons have also an advantage over diffs: they ease visualization when there are several versions. However, the 'versions evolution graph' proposed by Minor and Magnusson (1993) is a more explicit history device than the colours of Stick-Ons. Furthermore, there is no natural ordering of hues, so users cannot immediately know the history of the Stick-Ons without looking at the hues sequence scale at the bottom right of the window. (Repeated use of the system will familiarize the user with this scale.)

Hypertext solutions to versioning are comparable to Stick-Ons when there are various versions for exactly the same piece of text. For instance, when several alternatives have been proposed, in hypertext the user can easily navigate the various nodes containing the versions, with roughly the same effort as ungluing Stick-Ons, except that Stick-Ons will show each alternative version in 'most recent first' sequence in the context of the remaining text. However, when the versioning overlaps or corresponds to portions of an already versioned piece of text, Stick-Ons are clearly advantageous.

Use of Stick-Ons in practice

Stick-Ons are the versioning device provided by the SHADOW experimental joint authoring system. The SHADOW system does not include tools appropriate for group members to decide on preferred versions. The corresponding negotiation and decision are left for a face-to-face meeting to be held after the options have been proposed. SHADOW is also unaware of formatting instructions.

The photocopy tool can be considered not only as a way to have a clean, final

piece of writing to be exported to a formatter, but also as a useful device during intermediate stages. If the users believe they have a consistent non-final paragraph or document but full of now unnecessary Stick-Ons, they can photocopy the text and save it for future reference. Afterwards, Stick-Ons can be placed on the photocopy itself. However, the current SHADOW implementation does not provide a way to keep track of these complete non-final documents. It must be mentioned that the 'global' versioning technique is often used in the field of software engineering to manage source code versions, e.g. in SCCS (Rochkind, 1975) and RCS (Tichy, 1985). A new SHADOW release incorporating features for global versioning is being planned.

The solutions provided by the current SHADOW for author's identification (if needed) are not completely satisfactory. This is clear when other Stick-On attributes, such as time of creation of the Stick-On, are called for. A more general solution is to have an *about* button at the lower end of the window so that information concerning a current Stick-On can be displayed if the button is chosen. Again, this feature is being planned for the next release of SHADOW.

Comprehensive future group experimentation with the system should validate our hypothesis that author identification is not the most important Stick-On attribute (several researchers have argued, in fact, for anonymous contributions). In single-user situations, of course, identification of the author is useless.

A practical worry is also the number of colours to be assigned to Stick-Ons. As mentioned, the versioning system assigns different colours to each successive Stick-On within each paragraph. Theoretically, this ensures that if two Stick-Ons are contiguous or overlapping, their colours will be different (Thorell and Smith, 1990, suggest that if the purpose is to differentiate information, any number of colours is acceptable as long as there are sufficient contrast differences). In practice, after a number of colours, individual users may start to have difficulties discriminating new colours from the previous ones. In SHADOW, after a user-defined number of Stick-Ons have been created for a paragraph, a message is displayed suggesting that the paragraph be photocopied, if possible. The present SHADOW has a palette of 20 colours, in which hue and saturation have been assigned to distinguish one colour from another and to ease text readability when using them for background under the black letters.

Conclusions

The main feature of the Stick-On proposal is its intuitiveness: users are not required to learn concepts such as versions, or manipulate complex tools. The idea of gluing and ungluing colour patches handled with mouse clicks is simple enough to allow immediate use by people accustomed to page editors.

It is interesting to note that the use of Stick-Ons does not prescribe clearly marked stages of divergent/convergent collaborative work, as contexts do. Co-authors may add a contribution at any time, provided a final decision has not been made concerning the article.

We have performed limited experimentation with regular computer users making synchronous use of SHADOW. There was no previous training in this system at all. The basic Stick-On functionality and the photocopying metaphor

were quickly mastered. After some trials and use of multiple Stick-Ons, there was some surprise from the users. Questions arose when they tried to build a partially overlapping Stick-On over a shadow (the case illustrated by Figure 5) and the system automatically enlarged the Stick-On. After a brief explanation, they got the idea and continued working. Similar questions were asked when attempting to edit a paragraph locked by another user; the explanation plus the visual lock (see Figure 2) seemed enough to satisfy the user. Finally, some users asked about the meaning of the colour ordering and whether they could choose the colours.

Initial trials also showed that for simple text editing tasks (addition, deletion, replacement) in a single-user environment, Stick-Ons and their operations of gluing and ungluing are conceptually simple and appropriate. This benefit is obtained in addition to the included versioning, which may be handy, even in this case of only one user. Examples of uses include the preparation of versions of one article for various audiences, to serve as reminders to the author him or herself of the latest changes made, and to hold options before choosing the best wording.

Further experimental work is needed to completely assess the usefulness of the Stick-On as a versioning tool, both for single users and in a collaborative setting. This research could also prove the conjecture that versioning is valuable to generate positive conflict within an authoring group (by allowing people to easily express various proposals) and reduce negative conflict (by sparing people the need to always criticize each other's work).

Acknowledgements

This work was partially supported by the Chilean Science and Technology Fund (FONDECYT) grant No. 1940269. Edgardo Fabres worked on the development of the SHADOW system. Bob Dailey and three anonymous referees gave useful comments about the manuscript.

References

Baeza-Yates, R.A., Fuller, D.A., Pino, J.A. and Goodman, S.E. (1995) 'Computing in Chile: the jaguar of the Pacific Rim?' *Commun. ACM* **38**, 9, 23–28

Borges, M. and Pino, J.A. (1994) 'Additions to the card metaphor for designing human-computer interfaces' in *Proc. 4th Ann. Workshop on Information Technologies and Systems (WITS '94)* (Vancouver, Canada) 243–251

Conklin, J. (1987) 'Hypertext: an introduction and survey' *IEEE Comput.* **20**, 9, 17–41

Delisle, N.M. and Schwartz, M.D. (1987) 'Contexts — a partitioning concept for hypertext' *ACM Trans. Office Inf. Systems* **5**, 2, 168–186

Ellis, C., Gibbs, S.J. and Rein, G.L. (1991) 'Groupware: some issues and experiences' *Commun. ACM* **34**, 1, 39–58

Fish, R.S., Kraut, R.E., Leland, M.D.P. and Cohen, M. (1988) 'Quilt: A collaborative tool for cooperative writing' in *Proc. COIS '88: Conf. Office Information Systems*, ACM Press, 30–37

Irish, P.M. and Trigg, R.H. (1989) 'Supporting collaboration in hypermedia: Issues and experiences' in **Barrett, E. (ed)** *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information* MIT Press, 90–106

Minor, S. and Magnusson, B. (1993) 'A model for semi (a)synchronous collaborative

editing' in *Proc. ECSCW '93: 3rd European Conf. Computer Supported Cooperative Work*, (Milano, Italy), 219–231

Mitchell, A., Posner, I. and Baecker, R. (1995) 'Learning to write together using groupware' in *Proc. ACM CHI 95: Conf. Human Factors in Computing Systems* ACM Press, 288–295

Neuwirth, C.M., Chandhok, R., Kaufer, D.S., Erion, P., Morris, J. and Miller, D. (1992) 'Flexible diff-ing in a collaborative writing system' in **Turner J. and Kraut, R. (eds)** *Proc. CSCW '92 of the Conf. Computer-Supported Cooperative Work* (Toronto, Canada), 147–154

Neuwirth, C.M., Kaufer, D.S., Chandhok, R. and Morris, J.H. (1990) 'Issues in the design of computer support for co-authoring and commenting' **Halasz, F. (ed)** *Proc. CSCW '90: Conf. Computer-Supported Cooperative Work* (Los Angeles, CA, USA) 183–195

Posner, I.R. and Baecker, R.M. (1993) 'How people write together' in **Baecker, R.M. (ed)** *Readings in Groupware and Computer-Supported Cooperative Work — Assisting Human–Human Collaboration* Morgan-Kaufman, 239–250

Rochkind, M.J. (1975) 'The source code control system' *IEEE Trans. Softw. Eng.* **1**, 4, 364–370

Sharples, M., Goodlet, J.S., Beck, E.E., Wood, C.C., Easterbrook, S.M. and Plowman, L. (1993) 'Research issues in the study of computer supported collaborative writing' in **Sharples, M. (ed)** *Computer Supported Collaborative Writing* Springer-Verlag, 9–28

Shipman III, F.M., Marshall, C.C. and Moran T. (1995) 'Finding and using implicit structure in human-organized spatial layouts of information' in *Proc. ACM CHI 95: Conference on Human Factors in Computing Systems* ACM Press, 346–353

Thorell, L.G. and Smith, W.J. (1990) *Using Computer Color Effectively — An Illustrated Reference* Prentice-Hall

Tichy, W.F. (1985) 'RCS — a system for version control' *Software — Practice and Experience* **15**, 7, 637–654