Word processors now make it possible for many authors to work on the same document concurrently. But what can they actually do?

**BY RICARDO OLENEWA, GARY M. OLSON,
JUDITH S. OLSON, AND DANIEL M. RUSSELL**

Now That We Can Write *Simultaneously*, How Do We Use That to Our Advantage?

MODERN WORD PROCESSORS, like Microsoft Word OneDrive, SharePoint, and Google Docs, allow people to work on the same document at the same time. While systems that allow simultaneous writing have been demonstrated in research labs for some time, only relatively recently have such systems been available commercially for widespread use. Google Docs, for example, enables people to be at the same place in the document while adding, deleting, and moving text at will. Google Docs is very popular (for example, there are over 60 million users of Google Apps for Education worldwide and two million businesses use Google Apps for Work) and many people see the simultaneous writing feature as a great asset.

For example, when we analyzed in detail the writing patterns of undergraduate students enrolled in an advanced university course on project management, 95% of the documents exhibited simultaneous writing. Three documents were *only* written simultaneously.⁹

We, the four co-authors of this article, have experienced numerous sessions where simultaneous writing created notable benefits. Given that we can work simultaneously now, how can we harness that capability to make the work more efficient? What can you do with simultaneous writing? When might you *not* want to write simultaneously?

To answer these questions, we collected our stories, grouped them, and noted patterns across them. Each of the stories is told with the voice of one of the authors except the last, which is co-written by all four of us because it reflects how we wrote *this* article. Subsequently, we note the similarities and differences among our stories, and with “plural unity” created the scheme of six patterns and two epiphenomena. We believe the stories will serve as an inspiration to readers to work in new, beneficial ways.

Google Docs is relatively new and although some of our stories refer to the use of Docs, a number of the stories involve research prototypes or early commercial systems, some going as far back as the 1980s. Before we get to the stories, we first provide brief descriptions of the characteristics of each of these systems, listed in order of creation.

The Systems We Refer To

IDE, for Instructional Design Environment,¹⁰ was a tool to create instructional content by helping designers organize and create materials rapidly. *IDE* was built on top of the Notecards hypermedia system.^{5,6} *IDE* featured “cards” with different kinds of links between sections of text. While *IDE* was not originally intended as a multi-person collaborative system, people quickly realized they could partition work among a number of writers and get things done more quickly by working on different cards simultaneously. When someone

was working on a card, it was locked to the others. While this is technically not the kind of simultaneous editing that other tools have, the story where this was used for parallel simultaneous editing has some valuable lessons.

ShrEdit was a collaborative writing tool built in 1990.^{7,8} A shared document was hosted on a server and individuals accessed it from client machines on the same local network. The architecture allowed multiple people to write in the same document at the same time. ShrEdit had “selection locking,” meaning that one could write simultaneously within one character of one another.

Aspects was a commercial product available in the 1990s.⁴ Like ShrEdit, people could edit at the same time within one character of each other. Whereas ShrEdit supported only text, Aspects supported spreadsheets, drawings, and presentations in addition to text, similar to Google Apps but run-

ning on a local network and only available for Macintoshes.

Centra Symposium was a commercial system that allowed for audio and video conferencing as well as sharing of an object, like a document or a presentation. One could allow others to edit the shared object.

Google Docs is a commercial, multi-user, shared document editing system provided by Google beginning in 2006. Docs lets multiple people edit a document simultaneously, view the revision history of the document, and share the doc with specific people or broadly on the open internet.

The Power of Simultaneous Writing

We now move to telling our stories about how the power of simultaneous writing creates significantly more productive work settings. The stories that follow are grouped into five categories:

- Writing large documents (Stories 1–3)

- Writing short documents for class assignments (Story 4)

- Displaying and collaboratively creating meeting minutes (Stories 5–6)

- Teaching through shared documents (Story 7)

- Writing this document. (Story 8)

Story 1: Doc Build-It using Google Docs, told by Ricardo. To be useful to others, software needs to be documented. Many times software needs to be rewritten from scratch because the original could not be found or once found, could not be understood. Unfortunately, many software developers hate to document. Although technical writers, like me, can do the job, there are too few of us, and the process of understanding a technology and writing appropriate documentation can take years. Consequently, only core technologies get documented.

To solve this problem, I developed a documentation method called the Doc Build-It. A Doc Build-It is a single-

day event where a small group of engineers gather to simultaneously write documentation about a specific piece of technology. The Doc Build-It is a constrained activity, carefully designed to enable engineers to impart their expertise in a way that seems natural to them.

The Doc Build-It has three phases: preparation, composition, and editing. During the preparation phase, the writer meets with the technical lead to get a detailed overview of the technology to be documented. The writer then generates a prospective outline for the final document. The outline is granular to the point where each topic has two to three bullet points that suggest what content should appear in the topic.

The composition phase is a single-day event. The writer asks the technical lead to invite three to five key engineers from the team to the event. At the event, the writer first asks the engineers to give a verbal explanation of the technology to get them into a teaching mindset, a mindset that fits documentation. The writer then adjusts the prospective outline based on the explanation provided. Once the Build-It team agrees on the outline, the writer invites the engineers to claim responsibility for the topics in which they feel they have the most expertise. Their task is only to capture the ideas in the words that were spoken during the explanation phase. They then all write simultaneously in the same document. The fact that they can see each other as they produce the sections allows them to align their styles (like level of detail), make cross-references and double check for accuracy.

The writer encourages the engineers to avoid concerns about diction, spelling, punctuation, grammar, or structure. Their contribution to the document is their knowledge. This session normally takes about three hours for relatively simple topics, and up to seven hours for deeper topics.

After the composition phase, the technical writer alone polishes the document. Because the editing process can introduce semantic errors, the writer circulates the edited document for review with the engineering team.

The Build-It method had an enormous impact on productivity. The cost for one Build-It that I ran was an order

of magnitude smaller than a traditional documentation approach costing an estimated \$1,900 instead of \$18,000. Other Build-Its produce similar results, and had ancillary benefits like the identification of subtle bugs in the technology. These discoveries are possible because the event captures the attention of the most knowledgeable people about a topic, who focus closely on the design of the software in the process of creating a document.

Insights. Doing work in clear phases clarified the roles and simplified the whole process. The technical writer created an outline; the outline was then edited after the explanation. Then experts wrote down as much of their knowledge as they could. This knowledge was then cleaned up by the technical writer and reviewed for accuracy. A number of the following Stories have similar mixes of asynchronous and synchronous work.

Story 2: Writing a textbook using IDE, told by Dan. During the summer of 1988, I organized a team of 10 graduate students from the Stanford School of Education to help write a high school algebra textbook, fulfilling a requirement to have experience in designing coursework. The goal was to create a complete textbook with a team in less than 10 weeks. The team wrote this text using IDE.

Of the 10 students, two acted as lead designers and created an initial outline, assigning each chapter to a student author to create. To “write” a chapter, the author had to create a network of nodes in IDE, each linked to other nodes signifying whether the linked-to nodes were abstract concepts, conceptual details, refinements, practice problems or dependencies.

During writing, the authors had to identify any prerequisites that they needed to assume while writing their material. For example, the chapter on Trigonometric functions relied on the Pythagorean theorem to be introduced in an earlier chapter, written by a different author. For that dependency, an explicit prerequisite link would be created to signal that this concept (in one chapter) depended on another concept in another chapter (usually preceding).

The resulting network could then be graph-walked (by following links of a specified type) to produce a single,

linear, book-like output, incorporating diagrams and text from each node.

Every Monday, we would meet to review the work done during the preceding week. The team discussed and resolved any overarching issues, and clarified interactions between sections. Since each author could work more-or-less independently (except when they had to edit a node together sitting side-by-side), issues could be resolved fairly easily. The most common issues the team needed to resolve were questions about style, tone, language, and the framing of concepts.

Before each weekly meeting, we would print the graph of all the work done-to-date and post it on the wall for the entire group to see. This poster showed the week-by-week work done by each team member and the links between sections and various subparts. Not only was the graph useful for seeing where issues might arise, but it was also a powerful motivator for other team members to keep up.

During any single week, the authors could work independently in parallel, writing their nodes and creating any substructures as needed. When they discovered new prerequisites, they were responsible for creating those explicit links between the nodes as necessary. Of course, other authors could create new prerequisite links into their territory as well. For the cross-links, authors sometimes chose to work together side-by-side synchronously with one acting as scribe to the discussion.

After eight weeks, the project was completed—all the chapters had complete text, and all the cross-section prerequisite issues had been solved. The printout resulted in a 220-page textbook.

Insights. The book was primarily written in parallel, solo-author sessions, with some short, key moments of simultaneous work when issues arose and resolutions needed to be reached. The book was created over the course of eight weeks, contrasting with the Build-It which had preparation work, one long day of dumping material, and a few days of cleanup. Atlas, O'Reilly (atlas.oreilly.com) and BookSprints (www.booksprints.net) support a process similar to this story.

This story is also similar to the one told by Boellstorff et al.,² where four authors detail the creation of their coau-

thored book, written in Google Docs. It recounts initial hand-offs of responsibilities and sections to write. Later they capitalized on the ability to write simultaneously, to find who is writing and where they are writing, and initiate a conversation either by voice or associated chat feature. This was useful, they said, for both quick fixes and encouragement. They also report sessions where one typed while another was dictating, with a third coming along closely behind to make small edits, similar to the upcoming Story 5 about meeting minutes.

Story 3: Creating a committee report using ShrEdit, told by Judy. The computer science department at a major university has an external review every few years. An advisory committee of about eight people from outside the university comes to campus for a few days to hear what the various researchers are doing, review the curriculum, hear statistics on admissions, placement, and so on. Normally, after two-days of presentations, the committee would plan how they were going to write up their report, writing their parts, then reading and commenting on others asynchronously over the next four weeks.

Having found ShrEdit useful in our own work, we organized something like a Doc Build-It, but with some interesting differences. We brought the advisory committee to a special room that was set up with a large number of computers.

We had a single ShrEdit document open with an outline that I had written, which reflected the topics that had been presented over two days prior. All eight people began to write their reactions to and evaluation of the topics. The committee members read the input of others and added their own input. They chose to write wherever they wished, often adding to others' writing, and occasionally having text-based debates.

The committee worked for over an hour, all typing simultaneously. They produced an 11-page document; rough in its style, but full of good content. At the end, one committee member asked, "Where is the cleanup button?" All had a much-needed laugh. The leader of the team volunteered to take this rough draft and make it into good text, taking on a role very much like Ricardo's in the Build-It story. The

leader was grateful for the volume of raw material on which to build. This material was far richer than the minutes of a discussion would have been. And when they traveled home, all members of the committee, except the leader, were done.

Insights. Of interest in this story is the fact the writing was not divide-and-conquer like the two previous stories, but rather what we might call a "swarm," for the large number of co-authors involved in this process without a structured process. Everyone contributed to each of the sections at will, sometimes typing very close to another person's current entry. Except for the occasional laugh, the room was silent for an hour with only the sounds of keys clicking softly.

Story 4: Students writing assignments in class using Google Docs, told by Judy. My students in a Project Management course worked in groups to do a small project during the academic quarter. They had to turn in various documents that were common in formal project management practice, such as a business case, a scope statement, and so on. The students were required to use Google Docs and share their final document with me for grading.

With their permission, we analyzed three years worth of the documents—96 in all. We examined their

collaborative writing styles and work patterns, and correlated some key features with the quality of the final document. Work patterns were revealed through a tool, DocuViz¹¹ that shows a picture of the revision history, with slices in time of who wrote what when. The accompanying figure shows the DocuViz timeline visualization of one group's style. Students are represented with different colors, with the size of the stripe indicating length of contribution and different slices indicating when in time they were produced. This team had a lot of simultaneous work near the end.

Some 95% of those documents showed evidence of simultaneous writing, where we defined "simultaneous" as writing activity within seven minutes of the last action without closing and opening the document. One might expect these sessions of simultaneous work to be a "divide and conquer" style. While nearly one-third were, a large number of them showed editing by several people in the same paragraph or even in the same cell of a table.

Insights. We have less detail in this story about how the students managed to create, edit, and vet their work. But traces in the revision histories show not only explicit project management of assigning people to sections, but also freeform editing of their own en-

The DocuViz visualization of a team showing a session of simultaneous work near the end.

tries and, importantly, those of others. There was also a great deal of simultaneous work.

Story 5: Displaying and collaboratively creating meeting minutes using ShrEdit, told by Gary. In addition to using ShrEdit to conduct research,⁸ I used it to take minutes during research meetings. In these meetings, the ShrEdit document was shared and open on everyone's workstation, and we often projected it on a large screen at the front of the room. One person, usually me, took the lead, but everyone could edit.

Meeting participants would commonly fix typos and other errors (like correcting the spelling of someone's name) that I or another scribe created while typing rapidly. We often commented that the document looked as if Pac-Man was nibbling along behind the note taker. The shared view of the notes (either from the projected view or the individuals' views on their own computers) helped to maintain the group's focus, and make sure the points were captured correctly. The tool enabled efficient meetings and more accurate notes.

Normally, scribes for a meeting are so busy they cannot contribute easily to the conversation. But when we used ShrEdit, someone else took over the scribe role while the main scribe spoke. This allowed the main scribe to be a full contributor to the meeting.

In one important meeting with corporate sponsors, we projected the meeting minutes on a screen visible to everyone in the room. There were nine agenda items to cover in the afternoon-long meeting. After several hours of talking and taking copious notes, the meeting coordinator noted that we had covered only the first two items with seven still left to go. We had to move on. One of the sponsors said that he was not involved in any of the remaining items, but he had much more to say about the current topic. Consequently, the rest of us went on with the meeting while he continued to write in the appropriate section, hidden from our view. He titled his section "You haven't seen this yet." He wrote his thoughts for an hour, being able to refer to earlier parts of the minutes, while the rest of us finished our agenda. His time was not wasted, and we completed our task. The fact that this was a single doc-

ument (not an add-on of his thoughts from another document) made his additions easier to read in context.

Insights: Like Stories 1-4, the shared document had a prepared form (the agenda). While three of the previous stories divided the work and gave participants specific assignments to accomplish, in Story 5, one person was the main scribe. Other participants could silently and in parallel, add detail, correct errors, and in this case, continue to contribute while other meeting members went on with agenda items he wasn't involved in. Time was not wasted.

Story 6: Meeting minutes collaboratively and remotely created in Google Docs, told by Gary. The ACM SIGCHI Executive Committee has periodic meetings that last 2-3 days. An agenda is circulated in advance, and modified via email as new items arise. We use Google Docs to take minutes, the document seeded by pasting in the agenda from email. The agenda is often reorganized and modified on the fly.

Like the previous story, one member is designated as the main scribe, but any of us can edit at any time, and when the scribe speaks, someone else temporarily takes over his role.

Occasionally, some members attend parts or even the entire meeting remotely. We would often call remote members for specific items on the agenda, but they observed other parts of the meeting as well. Some even joined *without* having an audio or video connection. In this case, the Google Doc served as a kind of poor man's conferencing tool. The very low rate of change in the Doc works well for a remote participant who can then task switch at appropriate times. We noted also that non-native English speakers benefit from the minutes being a form of "closed captioning."

Often a member of the committee uses PowerPoint slides during their report; these are almost always pasted directly into the shared Google Doc. Occasionally, other kinds of materials are pasted into the Doc, such as links to some web material relevant to what we are discussing.

At another regular research group meeting, the minutes are never deleted but just pushed down, with the new agenda and notes appearing at

the top. This archive has proven useful in a number of occasions where someone could resurrect a forgotten "tabled" item.

Insights. This story shares the collaborative note taking with Story 5, but adds the feature of using the emerging document as low bandwidth conferencing and closed captioning.

Story 7: Teaching through noticing things in simultaneous work using Aspects, told by Judy. A former doctoral student of mine, who had graduated recently and was now employed at another university, was scheduled to give a presentation of our joint work at an upcoming conference. I asked the former student to prepare his talk in Aspects' presentation tool, and share it with me.

In a session in which both of us could see the presentation while speaking over the phone, the former student gave his presentation uninterrupted for timing purposes. At the end, I gave general feedback about the timing and order of things. I also suggested including different figures than those he had chosen. While he searched for the new figures and replaced them in the presentation, I made some small formatting changes on a number of slides starting near the end of the deck. When the former student returned, having completed the replacement of figures, he said, "Oh, I see what you're doing." He then changed earlier slides to fit the style of the changes that I was making. We discussed a few more changes in wording. When we were both satisfied with the result, he gave the presentation again without interruption. After two more small changes, we were done.

Gary had a similar experience using Centra screen sharing with the ability to write simultaneously. They went over a draft presentation, discussed it, and made changes in real time. The shared view focused the discussion; simultaneous editing sped the revisions.

Insights. What would have normally taken hours or days of back and forth took one hour. Modeling the types of changes that I wanted enabled the former student to "get it" without discussion. There could have been objections and discussion, but in this case he just copied my style changes without either.

Story 8: Writing this document together with Google Docs: How we did

it. Having discovered that all four of us had stories about writing simultaneously, Ricardo suggested that we write an article on this topic using Build-It. We all happily agreed, wanting both to write this article and to experience Build-It. We scheduled a day for the writing about a month after this initial meeting, with three of us collocated and Dan joining remotely.

We had a videoconference in advance of this day, where we discussed the general framework of the article and the kinds of stories we would use. This was all captured in a Google Doc that we co-wrote. Ricardo created a tentative outline based on this discussion. When we were together, in the first hour we discussed ideas and directions. Initially we were hampered by not having a clear idea of what the article was going to be. Build-It is designed for a divide-and-conquer strategy, not for the stage where we have to discover what we want to say. The outline Ricardo came in with did not capture the discussion we were developing, so we abandoned it.

Eventually we decided the best next step was to simply write up the set of stories that would serve as the raw material. We could read while others wrote. We often read something someone else had written and went back to add or change our own contribution. When everyone was finished, we then read them all in context which then led us to discuss a better organization for the article.

Indeed, in both writing the details and then reading each other's stories, we noted some similarities and differences. We made a working table directly in the document, to compare and contrast the stories. We hoped this would help in both ordering the stories in a sensible way and then supporting the development of the Discussion section.

In the course of creating this table, each of us recognized that we had left something out of the stories. Having noticed these emergent similarities, we combed our own memories for more examples. We named these gathered extra points "epiphenomena."

After about six hours of talking, writing and talking again, we were exhausted, but we had to decide on our next steps. Someone had to do the first pass of ordering the stories, making them a bit more similar in style, and then writ-

The writer encourages the engineers to avoid concerns about diction, spelling, punctuation, grammar, or structure. Their contribution to the document is their knowledge.

ing a draft of the Discussion using the table as a guide. Judy volunteered.

This next stage required someone to "own" the document temporarily, without others editing, because a reorganization and a first draft of a discussion required understanding the whole document to make the appropriate connections. When Judy finished, she alerted the others. But we didn't want everyone to edit at once. We first tagged Ricardo, a technical writer, to perform a more fine-grained copyedit. He tried to harmonize the voice and pace of the writing and identify or resolve features of the document that seemed to break the flow.

We then had a videoconference to talk through several important issues, making a list of things to do. We asked Judy and Gary to take the first pass at their parts of this list and then have Ricardo and Dan edit or comment further. Again, noting who was in control was important.

We had not talked about how we were going to manage edits, whether we would just make the changes and count on the revision history to undo things that we disagreed with, use Docs "suggest" mode, which is similar to Word's tracking changes, or use comments to discuss and suggest changes. Judy and Ricardo made changes directly. Dan then used comments to say, for example, "This needs to be spelled out." Dan later clarified that he did not want to edit someone else's work. These examples show that people come to the document with different ideas as to who has the authority or responsibility to change things. As Birnholtz and Ibara¹ found, making changes to others' writing is a social act with consequences having to do with trust and relationships.

The final stage of this collaboration involved putting the Google Doc into Word to fit the required two-column format that this journal requires. From there the editing was entirely hand-off with clear mention of responsibilities and time lines.

Insights. Unlike Build-It, the outline for this article emerged *after* the stories were written down, shared, and discussed. This article also involved using the document as a "holding place" to keep things (like the table) that might help the writing, or material that was

eventually deleted. There were times when simultaneous divide-and-conquer was appropriate; there were times when one person had to be in charge while she captured the organization of the emerging article; and there were times when serial hand-off for editing was appropriate. We realized that we should have explicitly discussed whether we would just edit, suggest, or comment for changes.

Discussion

The ability to write simultaneously in a shared document is a powerful advancement in technology. However, the literature has said little about the social process that harnesses the technological advancement of simultaneous writing for real benefit to the users. These stories attempt to shed light on this social process.

In almost all of these stories, someone led the effort by making some sort of structure: the tree in IDE, the outline in ShrEdit and Docs, the agenda in meetings, the presentation draft in Aspects and Centra. The one exception was the writing of this article. We had that discussion after we wrote our stories and read each other's. The structure emerged.

Writing simultaneously offers several benefits, including productivity gains, a deeper sense of satisfaction for time well spent, and practical training by imitation of a collaborator's style. On a tactical level, participants can move quickly toward a quality document because participants can see and emulate what others are doing. As people join in writing, they can view recent work in order to make their contributions fit the overall vision. The ability to work simultaneously on meeting minutes has benefits beyond the recording and correcting the content. Everyone was "on the same page."

Of course, not all collaborations may benefit from simultaneous work. There are sensitivities when someone changes your writing. There are sensitivities when others can see your process of writing (for example, if you are slow or a bad speller). Some may find it distracting to see the edits of others while they are writing or editing. Working simultaneously is not appropriate if you mistrust your colleagues, either because it is a competitive at-

When we examine the stories, it is striking that there is not more diversity in the experiences, but rather just a few different patterns of writing together.

mosphere or their expertise and abilities are not on par with your own. In some competitive cultures, there can be sabotage. In addition, many of the sessions of simultaneous work that we outline were for rough work, not for the final draft (although Story 7 contained the collaborative reformatting of a final presentation).

And, there can be technical difficulties, such as edit wars because with direct editing, one does not see what something was changed from. Also, if two people are in exactly the same point with one adding and one deleting, it can get very confusing. And, to write simultaneously, one has to be on a network, having to rely on server connectivity. In addition, if using a service in the cloud, some may have concerns about privacy.

Patterns of using simultaneous writing. When we examine the stories, it is striking that there was not more diversity in the experiences, but rather just a few different patterns of writing together. The stories cluster into two sets: Four patterns of simultaneous work and two patterns of accompanying asynchronous hand-offs.

The first four stories and the account of writing this article employ a simultaneous *divide-and-conquer* strategy. At some points in the creation of the document, all co-authors wrote at the same time. Most often they wrote in different sections of the document.

Stories 5 and 6 employ a main scribe, with a second or third scribe involved either to immediately take over when the primary scribe speaks or to do ancillary additions or edits. We call this the *rotating scribe* pattern.

Story 5 employs a *branching* pattern, where when one person is not involved in the immediate conversation, they use the time productively to write more for others to read later. In essence, the one person is creating a new branch in the minutes, while the others proceed. This pattern is a variant of the divide-and-conquer.

The fourth pattern is exemplified in Story 3, what we call the *swarm*. In this pattern, everyone is in the document writing their parts, reading other's and commenting or correcting them. No one is assigned a section; they all are responsible for the whole document. This is also similar to the teaching sto-

ries, where the “teacher” exemplifies what the “student” is to emulate, and they collectively finish the document.

The fifth pattern, the *cleanup*, is a solo kind of work that appears in many of the stories. For example, in our own writing, there was one big simultaneous session and then Judy took over to both re-organize the stories into clusters and write the text that represented the discussion captured in the table. In the committee report, the chair took on the role of cleaning up the rough text and synthesizing content. In BuildIt, Ricardo takes on the responsibility of cleaning up the text and coordinating tone, turning rough text into good prose.

The sixth pattern, again often represented in many of the stories, is the *hand-off*, where different authors are “in charge” for periods of time for re-organization or fact-checking and simpler editing. This often accompanied sessions of simultaneous work.

We also note that collocation for synchronous writing was important but not necessary. Collocation provides immediate access to other participants for things such as clarification, seeing people’s expressions, indicating that one wanted a turn to speak, and so on. However, many of the stories had successful contributions from remote people, even without audio or video connectivity. In one case, remote participants *only* saw the evolving Google Doc. For some purposes, this level of participation was sufficient.

Epiphenomena. Late in our discussion of the similarities and differences among the stories, we noted some epiphenomena—unusual behavior generated by the fact the work was being created synchronously in view of all.

One epiphenomenon involved *humor*, often an important social component of intense work. For example, as the scribe in meetings where the minutes are projected, Gary would often type slightly snide comments about what was being said and quickly erase them. For example, if someone was talking too long, Gary would write in big letters but out of the speaker’s view, “Is it time for lunch yet?” and then quickly erase it.

A student in one Project Management team, deep in discussion in a long simultaneous-editing session about their client, pasted in a funny

picture of a grizzly man wearing a bear hat and the text “this guy!” The picture was in the document for about a minute then deleted. The Visiting Committee read emerging debates inside the report, eliciting giggles occasionally. At the end, one member of the committee also remarked for humor, “Where is the Clean-It-Up button?” The collective laugh was one of appreciation and relief that the session was over.

A second epiphenomenon we noted is that visibility is motivating. In using IDE to write the textbook, the visible presentation of progress on the book motivated people. Similarly, while the document is live and worked on simultaneously, one can see where the activity is and be motivated to read carefully and talk about any arising issues either through text chat or voice conversation. When there is visible activity, people feel compelled to focus on the document being created. One attends to “seismic activity.” We believe the student who pasted in a funny picture during a simultaneous working session used the humor to induce motivation to keep working. One student in another team pasted in a picture of a shovel and wrote “Work Hard!”

Conclusion

Writing simultaneously is an extremely powerful capability, now widely available in commercial software. It is often successfully mixed with some hand-off writing and some sessions where one person takes charge to integrate the material and voice. But technology alone does not make enhanced productivity and satisfaction, people do. What we offer here are a number of stories and commentary on what makes this kind of writing so powerful.

We follow the stories with six patterns of writing when simultaneous work is possible. Team members now need to plan the style of work (some of which will include simultaneous writing) that would fit the kinds of goals at hand. As outlined by Glushko³ some of the overall collaborations will be preplanned (what he calls hierarchical collaboration), co-developed by the authors (what he calls consensus collaboration), or a free-for-all (what he calls open collaboration). Each approach has its strengths and weak-

nesses, so choosing appropriately for the task at hand is important. And, as mentioned earlier, not all teams and occasions will benefit; cooperation and trust are essential.

We have provided a set of examples where we have worked in new ways that are very productive, and therefore satisfying. The next step is for you to analyze the collaborative writing situations you are in day-by-day and craft a method that best suits. The possibilities are very rich.

Acknowledgments. Portions of this work were supported by National Science Foundation grant ACI-1322304 and a Google Focused Research Award to Judy and Gary Olson. Helpful comments on an earlier draft were provided by Tom Boellstorff, Bonnie Nardi, and Bob Glushko along with anonymous reviewers. C

References

1. Birnholtz, J. and Ibara, S. Tracking changes in collaborative writing: Edits, visibility, and group maintenance. In *Proceedings of CSCW'12*, 2012, 809–818.
2. Boellstorff, T., Nardi, B., Pearce, C., and Taylor, T.L. Words with friends: Writing Collaboratively Online. *Interactions* 20, 5 (Sept.-Oct. 2013), 58–61.
3. Glushko, R. Collaborative authoring, evolution, and personalization for a ‘Transdisciplinary’ textbook. In *Proceedings of OpenSim 2015*.
4. Group Technologies. Aspects: The first simultaneous conference software for the Macintosh, Version 1. Manual, (1991). Group Technologies, Inc., Arlington, VA.
5. Halasz, F.G., Moran, T.P., and Trigg, R.H. Notecards in a nutshell. In *Proceedings of the SIGCHI-GI Conference on Human Factors in Computing Systems and Graphics Interface* (1987), 45–52.
6. Irish, P.M., and Trigg, R.H. Supporting collaboration in hypermedia: Issues and experiences. *J. American Society for Information Science* 40, 3 (1989), 192–199.
7. McGuffin, L. and Olson, G.M. ShrEdit: A shared electronic workspace. CSMIL Tech. Report, 1992. The University of Michigan, Ann Arbor, MI.
8. Olson, J.S., Olson, G.M., Storrosten, M. and Carter, M. Groupwork Close Up: A Comparison of the Group Design Process With and Without a Simple Group Editor. *ACM Trans. on Information Systems* 11, 4 (1993), 321–348.
9. Olson, J.S., Wang, D., Zhang, J. and Olson, G.M. How people write together now. *Trans. Computer-Human Interaction* 24, 1 (2017), 1–40.
10. Russell, D.M., Burton, R.R., Jordan, D.S., Jensen, A.-M., Rogers, R.A., and Cohen, J.R. Creating instruction with IDE: Tools for instructional designers. *Intelligent Tutoring Media* 1.1 (2009), 3–16.
11. Wang, D., Olson, J.S., Zhang, J., Nguyen, T. and Olson, G.M. DocuViz: Visualizing Collaborative Writing. In *Proceedings of the Conference on Human Factors in Computing Systems*, (2015), 1865–1874.

Ricardo Olenewa is a technical writer living in Waterloo, Ontario, Canada.

Gary M. Olson (gary.olson@uci.edu) is Professor Emeritus of Informatics at the University of California at Irvine.

Judith S. Olson (jolson@uci.edu) is Professor Emerita of Informatics at the University of California at Irvine.

Daniel M. Russell (drussell@google.com) is a senior research scientist at Google, Mountain View, CA.

Copyright held by owner(s)/author(s).