Author manuscript, published in "International Conference on Enterprise Information Systems and Web Technologies - EISWT-07 (2007) 176-182"

inria-00129222, version 1 - 20 Apr 2007

New Work Modes for Collaborative Writing

Hala Skaf-Molli, Claudia-Lavinia Ignat, Charbel Rahhal and Pascal Molli
LORIA-INRIA Lorraine, France
{*Hala.Skaf,Claudia.Ignat,Charbel.Rahhal, Pascal.Molli*}*@loria.fr*

Abstract

In the recent years, due to the emergence of new models of production based on collaboration, collaborative writing tools started to be increasingly used by various communities. Due to great variety of groupware software, there is the need of criteria for discriminating the working modes supported by these tools. In this paper we propose a new model for the classification of collaborative writing tools. Based on our model, we categorize most popular existing collaborative tools. We also propose novel working modes that are not adopted by existing tools and that improve the process of collaboration.

1. Introduction

Collaborative writing (CW) is becoming increasingly common, often compulsory in academic and corporate work. Most of all written work is produced collaboratively [5], writing journal papers, technical manuals and planning presentations being few examples of common collaborative writing activities.

Many definitions of CW exist [11]. In this paper we consider the most commonly used definition according to which *CW is the process of two or more people working together to create a complex document*.

The major benefits of collaborative writing include reducing task completion time, reducing errors, getting different viewpoints and skills, and obtaining an accurate text [20, 17]. On the other side, many challenges are raising, ranging from the technical challenges of maintaining consistency and awareness to the social challenges of supporting group activities and conventions across many different communities.

The nature of collaboration varies extensively [20] in terms of the group writing strategies, relationships and roles of the team members and the proximity and synchronicity of collaborative activities. For instance, for collaboratively writing a document various strategies exist: users can jointly write a document by working closely together or they can work separately, their work being subject to review by other group members. Relationships among users involved in the collaboration are either established by organisational policies of by users themselves when they decide to work together on a project. An example of an application where user relationships are not imposed by organisational policies is Wikipedia [21]. Various working modes exist depending on the degree of proximity and synchronicity of collaborative work: some collaborative groups all work in the same location and on the same time schedule, while other groups work on different schedules and may be located thousands of miles apart.

The CSCW Matrix [10] illustrated in figure 1 is usually used to categorize groupware software [6] according to the degree of physical proximity of the group members and the degree of synchronicity of writing activities (when the author writes). In spite of the popularity of this matrix in the CSCW community, it suffers from major drawbacks. Firstly, definitions of synchronous and asynchronous modes of communication are confusing. Secondly, this matrix is 19 years old today and it is not anymore discriminative for current collaborative writing tools. In what follows, we present in detail these drawbacks.

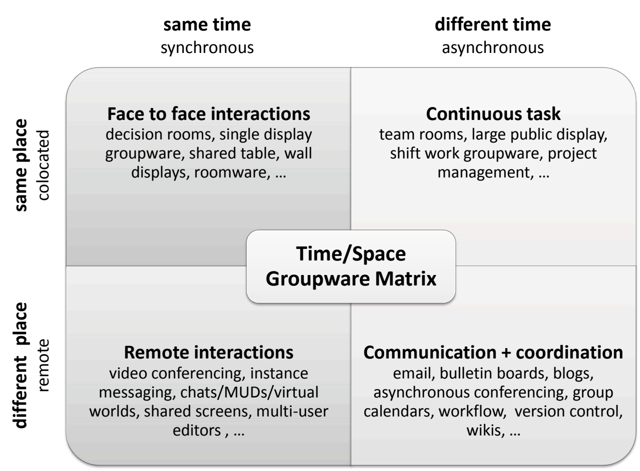

Figure 1. CSCW Matrix

Definitions of synchronous and asynchronous modes of interaction are confusing. For Noel [17] synchronous interaction means synchronous access, i.e. concurrent access. For Ellis [6] synchronous interaction means real-time interaction, and asynchronous interaction means non real-time interaction. Baecker [1] defines asynchronous groupware as supporting ”communication and problem solving among

inria-00129222, version 1 - 20 Apr 2007

groups of individuals who contribute at different times, and typically also are geographically dispersed".

Moreover, some existing collaborative tools can be classified as belonging to several categories represented by the CSCW matrix. For instance, it is the case of Version Control Systems (VCS) and Google Docs. VCS [7] use the copy-modify-merge paradigm. This allows users to work insulated and therefore not being bothered with modifications done by other authors. Insulated work is considered as an asynchronous mode of interaction although people often work in same time. Thus, according to the matrix, VCS should be also classified as synchronous (same time) systems. Recently, web-based collaborative tools for editing documents and spreadsheets developed by major software vendors such as Google (Google Docs and Spreadsheets [9]) attracted the interest of a large number of users. Documents can be shared, and can be opened and edited by multiple users at the same time. Google Docs is considered as an online real-time collaborative writing tool. However, Google Docs uses exactly the same principles as a VCS. The user can import the document from the server in order to edit it through the web interface. When users save their changes, the document is saved on Google's servers. As we previously shown, VCS can be considered both synchronous and asynchronous. Therefore, Google Docs can be also classified as both synchronous and asynchronous according to the CSCW matrix.

Moreover, multi-synchronous tools [4, 14, 16, 18] that incorporate synchronous and asynchronous work can fit in any classification of CSCW matrix. As we previously showed, this is also the case of Google Docs. However, multi-synchronous work supports insulated work while Google Docs does not. Therefore, two different classes of tools, multi-synchronous tools and Google Docs, are equally classified according to the CSCW matrix.

Finally, the matrix supposes that all group members work in the same working mode at any moment of time. However, this is not the most general case regarding the user working modes during a collaborative activity. At a certain time during the collaboration some group members can work in real-time while others prefer working in insulation.

We therefore see the need of new criteria for classification of various modes of collaboration. In this paper, we present a model for the categorization of interaction modes that offers support for the classification of existing tools. Moreover, in this paper we present novel modes of collaboration derived from our model, that are not currently supported by current tools. In [17], it was pointed out that new interaction modes can make collaborative writing more attractive for end users and we think that the novel working modes that we propose can improve the adoption of CW tools by a large public.

The paper is organized as follows. We start by presenting our new model for the classification of collaborative writing modes. This model is called SRI, SRI standing for Sending, Reception and Integration. It allows to distinguish between the different interaction phases during a cooperative writing, as we show in section 2. We then go on in section 3 to categorize existing collaborative systems according to our model. In section 4, we present new modes of interaction and we show their importance. In section 5, we present our concluding remarks and directions of future work.

2. The SRI model

In this section, we present our interaction model for a collaborative writing system.

A collaborative writing system consists of a set of participant systems connected by a communication network. In this paper,we are going to use the term site for denoting a participant system. There is one site per user (author). We consider that the shared document is replicated at each site.

In other words, a collaborative writing system is modelled as follows. It considers n sites, each site owns a copy of shared data. When a site performs an update, it generates a corresponding operation. An operation is processed in four steps:

1. Execution on one site,
2. (S)ending to other sites,
3. (R)eception by other sites,
4. (I)ntegration on other sites.

According to the phases of processing of an operation, sending, reception and integration, we called our model of interaction in a collaborative writing system as the SRI model.

When an operation is generated, it is immediately executed on the local site. On most systems, this means that its effects are immediately visible on the local site. The operation is then sent to the other sites, which receive and integrate it.

Most existing systems do not distinguish between the phases of reception and integration. In SRI, we consider that the two phases are different and that it is possible to receive operations without integrating them, as we show later in this paper.

In what follows we describe the phases of our model, i.e. sending, reception and integration and the parameters that describe these phases.

2.1. Sending

This phase raises three issues: what to send, when to send and to whom to send the data. Some pre-conditions have to be verified in order to send data. For example, in order to send operations, users have to be connected.

Pre-condition The sending phase can be refused if pre-conditions are not satisfied. Pre-conditions are specific

inria-00129222, version 1 - 20 Apr 2007

New modes	Sending	Reception	Integration	Pros	Cons
Single-user editor + Mail	User	User	Unsupported	Easy to use, Insulation, Offline mode	Preplanning required, No integration support
Real-time editor	Immediate	Immediate	Immediate	Reactive writing, No preplanning, No sending pre-condition	No insulation, No offline mode, Small size group
Google Docs	System / User triggered	System / User triggered	Immediate	Online editor, Reactive writing, No preplanning, No sending pre-condition	No insulation, No offline mode, Low responsiveness
Wiki	User (if up-to-date)	User	Manual	Online editor for web pages, Insulation	Sending pre-condition, Blind modifications, No offline mode, Poor integration support
CVS/Subversion	User (if up-to-date)	User	Immediate	Insulation, Offline mode	No real-time, Blind modifications, Risk of redundancy
SAMS	User / Immediate	User / Immediate	Immediate	Work modes transition	Blind modifications in UUI mode
NICE	User / Immediate / System -triggered	User / Immediate / System -triggered	Immediate	Work modes transition	Blind modifications in UUI mode

Figure 2. SRI Matrix

to each collaborative work mode. For example, in the copy-modify-merge paradigm, users cannot send operations if they are not up-to-date with last published copy. In turn taking, users cannot send data if they do not have the token.

What Sites send operations. For example, in a wiki system, when a user saves his page, the browser sends an operation containing the new page content. In VCS systems, when a user commits his changes, the system sends a patch containing all modifications.

When It is possible to send the operation immediately, user triggered or system triggered.

Immediately Operations can be sent immediately, such as in real-time writing software. When a user types a character, the operation is generated locally and broadcast immediately to the other users.

User Triggered Sending modifications can be deferred to a moment decided by the user due to one or more of the following reasons:

- to preserve users privacy,
- to temporarily allow users to work offline and publish their changes at a later time,
- to do not bother other users with very draft changes.

For example, insulated work in VCS tools implies User Triggered Send mode.

System Triggered The system itself sends automatically modifications based on a time interval, as in Google Docs for example.

to Whom It is possible to send changes to all participants or to some participants. In some situations, it is useful to make changes public only to certain participants. For example, it is the case that a user wants to send his draft just to a specific reviewer before making it publically available to other users. A VCS system can support this scenario by using the branching mechanism. The user

inria-00129222, version 1 - 20 Apr 2007

can then choose to commit on the branch shared with the reviewer.

2.2. Reception

Concerning the reception phase, the issues that are raised are when to enable the reception of operations and what to receive.

When As in the sending phase, it is possible to receive the modifications immediately, user triggered or system triggered.

Immediately Real-time editors such as SubEthaEdit [19] receive modifications as soon as possible.

User Triggered Other systems such as VCS let users choose when they want to receive remote changes.

System Triggered In Google Docs the system decides when to receive remote changes.

What A site can receive all modifications sent by the other sites or can filter what to receive. It is possible to filter according to the source or to the content. For example, in a VCS system such as Darcs [3], users can select patches they want to retrieve from remote sites.

Once remote changes are received, it is necessary to decide when these changes are integrated.

2.3. Integration

Concerning the integration phase, we analyse next when operations can be integrated and what operations can be integrated.

When It is possible to integrate operations immediately, in a user triggered or system triggered manner.

Immediately In most existing systems such as real-time groupware, Wikis and VCS, when remote data are received, integration is performed immediately.

User Triggered In systems such as Mercurial [12] or Darcs [3], remote operations can be received, but not integrated immediately. These systems allow users to inspect remote changes before their integration.

System Triggered To our knowledge, no existing system supports this mode of integration.

What Users can select what operations to integrate. For example, in CVS [2] users can choose which files to update.

After defining the SRI model, the next step is to classify existing collaborative tools with our new Model.

3. SRI Matrix

The SRI matrix classifies existing collaborative tools according to the criteria Sending, Reception and Integration defined by the SRI Model. Figure 2 presents a classification of the most representative collaborative writing tools according to the SRI model. The table lists the advantages and disadvantages of the interaction modes offered by the analysed tools.

In spite of the fact that many specialized collaborative writing systems have been developed in the last 30 years, many people continue to collaborate using a single text editor and the email. In this case, users decide when to send modifications. They receive remote modifications when reading their email and they manually perform integration. This work mode requires a good planification of activities [17]. Integration raises no problems if people work sequentially. If the document is well segmented, different people can work in parallel on different associated segments. For example, a conference proceeding contains a collection of disjoint papers and the integration of papers in the proceedings has to be performed manually, but is trivial process.

As we previously mentioned, a requirement of this work mode is a good planification. However, collaborative writing is often a non-linear dynamic process [8]. Therefore, this work mode is restricted to the collaborative writing mode where a document can be decomposed in disjoint segments. When document decomposition is not possible, the integration of overlapping changes is too complex to be done manually performed.

In real-time editors such as SubethaEdit [19] or Coword [22], operations are sent immediately and integrated as soon as they are received. Real-time editors allow reactive writing [11] where users react and adjust each others modifications and additions without a pre-planning and explicit coordination strategy. However, real-time editors do not support insulated work. Moreover, this work mode is only suitable for small size groups.

In Google Docs operations can be sent either automatically by the system at very short intervals or by users at certain moments decided by them. Similarly, reception of operations is triggered by users or automatically performed by the system after a default time interval. As soon as operations are received, they are integrated on the current version of the document. As in real-time editing, Google Docs adopts a reactive writing strategy and features all advantages mentioned for real-time editors. However, due to the small delay for sending and reception of operations, the group reactiveness is decreased compared to real-time editing tools.

Although according to our criteria of classification and evaluation, Google Docs has less positive features than real-time editors, it is a very popular tool. Its popularity raises from the tendency of big vendors to adopt web as workplace. The advantages of using an online editor are easy

inria-00129222, version 1 - 20 Apr 2007

New modes	Sending	Reception	Integration	Pros	Cons
UIU Mode	U	I	U	Insulation	Blind modifications, Preserve privacy
IIU Mode	I	I	U	No blind modifications, Insulation	No Privacy, Intermediate results

Figure 3. New SRI Modes

access and instant sharing of documents from everywhere using a standard web browser. Moreover, real-time editing systems still remain at the stage of research prototypes focused on one aspect of collaboration such as concurrency control and have not yet reached the maturity of commercial systems. Therefore, it is possible to build an online real-time editor, but currently, none is available.

The Wiki system is the first collaborative online editor for editing web pages. Ease of use, instant sharing and the possibility to edit pages from everywhere make wiki systems very popular. Wikis use the copy-modify-merge paradigm. Sending is allowed only if all concurrent changes are already received and integrated. Due to the copy-modify-merge paradigm used, Wikis allow insulated work. However, no awareness is provided during insulated work and therefore blind modifications can occur. The user will see concurrent changes only when saving the page. The integration mechanism simply warns users that concurrent changes exist, but users have to manually reconcile their version with the last version.

CVS/Subversion implements the copy-modify-merge paradigm and thus allows insulated work. Users decide when operations should be published. However, sending operations can be rejected by the system if user copies are not up to date. Users decide when to receive updates and once received, operations are immediately integrated. Therefore, these systems make no distinctions between the phases of reception and integration. Users can work offline, a network connection being only required for sending and receiving operations. The main disadvantage of the offline work mode is blind modifications. Blind modifications can lead to redundancy or to useless work. For example, a user can modify a section while another user deletes the same section, leading to useless work. Two users can modify the same section concurrently with the same intentions and, therefore, redundant work is performed.

SAMS [14] is a multi-synchronous editor that allows users to work in insulation or in real-time as well as supports the transitions from one mode to another. Operations can be sent immediately or triggered by users. Similarly, reception can be performed immediately or triggered by the users. The main advantage of multi-synchronous editors is the support they offer for switching to a work mode corresponding to user needs. Unfortunately, when users work insulated, as with CVS, blind modifications can be produced.

NICE [18] is an extension of SAMS as it offers more flexibility for the sending/receiving operations. Both SAMS and NICE systems do not make distinction between the reception and integration phases, this being the main reason that leads to blind modifications.

Unlike all existing tools, we point out that insulated work mode can be achieved by deferring integration while receiving operations in real-time. If concurrent operations are available, it is possible to provide awareness of concurrent changes in real-time, and therefore preventing blind modifications. In the next section we present two new work modes for collaborative writing that prevent blind modifications.

4. New SRI Modes

Compared to 20 years ago, today people are working most of the time connected. This situation creates the opportunity to have a new work mode where people work connected and insulated. In this mode users can receive remote operations in real-time and decide when to integrate them locally. Real-time reception of remote operations gives the opportunity to provide a new kind of awareness for collaborative editors. For example, a user working on a section can be aware in real-time that concurrent operations have been performed *on this section*. Figure 3 illustrates our two new modes of interaction.

The first new mode UIU is an improvement of the UUI mode offered by CVS. Instead of deferring reception as in CVS, we defer integration. Blind modifications can still occur, but it is possible to indicate in real-time the location of concurrent changes in the document as soon as this information is available. For example, in [15], authors propose an awareness system based on an editing profile that counts operations performed on different parts of the document, such as paragraphs, sentences and words. Since concurrent operations are available only when updates are performed, the editing profile is computed only after integration is performed. Our new work mode would allow to receive concurrent operations and compute the editing profile proposed in [15] in real-time.

The main disadvantage of the UIU work mode is blind modifications. In order to eliminate blind modifications, we introduced the IIU mode. In this mode, operations are sent and received immediately as in real-time editors. To preserve insulated work, integration is triggered by users. In

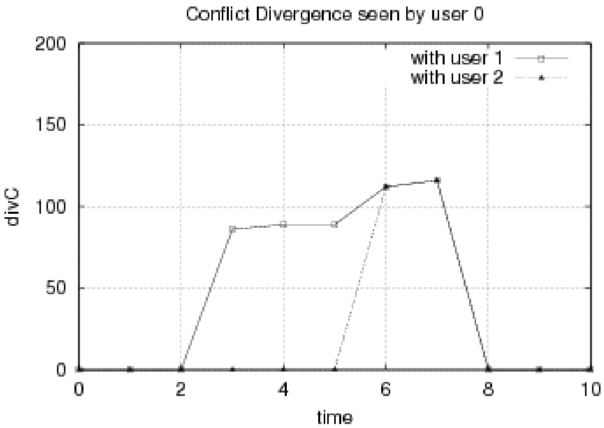

Figure 4. Divergence metric

the meantime, a new kind of awareness [13] can be used to notify users about the amount of concurrent changes and their location in the document. In [13] authors proposed to compute a divergence metric based on the counting of conflicting concurrent operations. They claim that if users can be aware of the amount of divergence in the system, they will start to communicate and generate some kind of auto-coordination to prevent expensive integration phases. They introduced a new widget to visualize divergence as illustrated in figure 4.

In this figure, a user can see the evolution of the amount of divergence with two other users. Our IIU new work mode allows to have such widget computed in real-time. On the other hand, it means that users agree to publish their modifications in real-time, their privacy being violated in this way. A mode where local operations are not sent immediately but are system triggered as in Google Docs can be proposed. This allows to find a balance between insulation and blind modifications. Solving privacy is currently a limitation of the IIU mode and we are currently investigating this issue.

5. Conclusion

This paper proposes novel criteria for the classification of various modes of collaborative interaction. It describes the SRI model which is based on the three phases of processing of a message exchanged between users during the collaboration, i.e. the phases of sending, reception and integration. It classifies existing collaborative tools by using SRI model and it shows the pros and cons of the working modes used by these systems. It also proposes novel working modes that are not adopted by existing tools and that improve the awareness of users while preserving their privacy by working in insulation.

We plan to investigate other new working modes for collaboration as well as implement some of the novel proposed modes in our systems [14].

References

[1] R. Baecker, J. Grudin, W. Buxton, and S. Greenberg. *Readings in Human Computer Interaction: Towards the Year 2000*. Morgan-Kaufman, 2nd edition edition, 1995.

[2] B. Berliner. CVS II: Parallelizing Software Development. In *Proceedings of the USENIX Winter Technical Conference*, pages 341–352, Washington, D. C., USA, January 1990. USENIX Association.

[3] Darcs. *Distributed. Interactive. Smart*. http://darcs.net/.

[4] P. Dourish. The Parting of the Ways: Divergence, Data Management and Collaborative Work. In *Proceedings of the European Conference on Computer-Supported Cooperative Work - ECSCW'95*, pages 215–230, Stockholm, Sweden, September 1995. Kluwer Academic Publishers.

[5] L. Ede and A. Lunsford. *Singular Text/Plural Authors: Perspectives on Collaborative Writing*. Southern Illinois University, 1990.

[6] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some Issues and Experiences. *Communications of the ACM*, 34(1):39–58, January 1991.

[7] J. Estublier, editor. *Software Configuration Management: Selected Papers of the ICSE SCM-4 and SCM-5 Workshops*, volume 1005 of *Lecture Notes in Computer Science*. Springer Verlag, October 1995.

[8] J. Galegher and R. E. Kraut. Computer-mediated Communication for Intellectual Teamwork: An Experiment in Group Writing. *Information Systems Research*, 5(2):110–138, 1994.

[9] Google docs & spreadsheets. *Create and share your work online*. http://docs.google.com.

[10] R. Johansen. *GroupWare: Computer Support for Business Teams*. The Free Press, New York, NY, USA, 1988.

[11] P. B. Lowry, A. Curtis, and M. R. Lowry. Building a Taxonomy and Nomenclature of Collaborative Writing to Improve Interdisciplinary Research and Pratice. *Journal of Business Communication*, 41(1):66–99, January 2004.

[12] Mercurial. *A fast and lightweight Source Control Management system*. http://www.selenic.com/mercurial/.

[13] P. Molli, H. Skaf-Molli, and G. Oster. Divergence Awareness for Virtual Team Through the Web. In *Proceedings of World Conference on the Integrated Design and Process Technology - IDPT 2002*, Pasadena, California, USA, June 2002. Society for Design and Process Science.

[14] P. Molli, H. Skaf-Molli, G. Oster, and S. Jourdain. SAMS: Synchronous, Asynchronous, Multi-Synchronous Environments. In *Proceedings of the Conference on Computer-Supported Cooperative Work in Design - CSCWD 2002*, pages 80–85, Rio de Janeiro, Brazil, September 2002.

[15] S. Papadopoulou, C.-L. Ignat, G. Oster, and M. Norrie. Increasing Awareness in Collaborative Authoring through Edit Profiling. In *Proceedings of the IEEE Conference on Collaborative Computing: Networking, Applications and Worksharing - CollaborateCom 2006*, pages 1–10, Atlanta, Georgia, USA, nov 2006. IEEE Computer Society.

inria-00129222, version 1 - 20 Apr 2007

[16] N. Preguiça, J. L. Martins, H. J. Domingos, and S. Duarte. Supporting Multi-Synchronous Groupware: Data Management Problems and a Solution. *International Journal of Co-operative Information Systems*, 15(2):229–258, June 2006.

[17] S. N. J.-M. Robert. Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work - JCSCW*, 13(1):63–89, 2004.

[18] H. Shen and C. Sun. Flexible Notification for Collaborative Systems. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work - CSCW'02*, pages 77–86, New Orleans, Louisiana, USA, November 2002. ACM Press.

[19] Subethaedit. *Collaborative text editing. Share and Enjoy*. http://www.codingmonkeys.de/subethaedit/.

[20] S. G. Tammaro, J. N. Mosier, N. C. Goodwin, and G. Spitz. Collaborative Writing Is Hard to Support: A Field Study of Collaborative Writing. *Computer-Supported Cooperative Work - JCSCW*, 6(1):19–51, 1997.

[21] Wikipedia. *The free encyclopedia that anyone can edit.* http://wikipedia.org/.

[22] S. Xia, D. Sun, C. Sun, D. Chen, and H. Shen. Leveraging Single-User Applications for Multi-User Collaboration: The CoWord Approach. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work - CSCW 2004*, pages 162–171, Chicago, Illinois, USA, November 2004. ACM Press.

inria-00129222, version 1 - 20 Apr 2007