Towards formal evaluation of collaborative work

Ricardo Baeza-Yates and **José A. Pino**
Depto. de Ciencias de la Computación, Universidad de Chile
Blanco Encalada 2120, Santiago, Chile

Abstract

Introduction. Computer Supported Cooperative Work (CSCW) projects are difficult to evaluate when implemented at most organizations. There are many variables and some of these are qualitative and hard to assess. However, there are other variables that could be measured and, thus, for a specific CSCW system, management could have a performance estimate.
Method. A groupware application is modelled, focusing on the work done and time spent on the collaboration.
Analysis. The following variables and their relations are studied: quality of the outcomes, number of people involved and time spent on the overall task, and total work done.
Results. An application - collaborative retrieval - is formalized to illustrate the model. For this application, a specific heuristic is proposed to the case when many people search for the same information, thereby increasing the recall and precision of the answer. The evaluation methodology is applied to this case, showing some experimental results.
Conclusion. We present an initial attempt to formally evaluate performance measures related to CSCW applications.

Introduction

The introduction of Computer Supported Cooperative Work (CSCW) tools within an organization can have a variety of positive outcomes (Eason and Olphert 1996). Generally speaking, these benefits can be classified into three broad categories:

1. benefits relating to an improved quality of the end product. Better documents, designs or decisions may be the result of the use of collaborative editors, computer-aided design or electronic meeting systems, respectively, when compared with similar outcomes obtained with computer applications intended for individual use.
2. those relating to the gains received by the individual members of the group and by the group as a whole. These may be in terms of knowledge acquisition, but mainly as improvements in social ties and friendly relationships. In one experience with CSCW, Ellis *et al.* (1991) mention the satisfaction of the group members with the result, which was a composite of individual contributions.
3. efficiency: to reduce wasted time or opportunities for

As an example of efficiency gains, consider the writing of an article by a group of co-authors. One simple non-CSCW approach consists of all co-authors seated around a table thinking about the article with only one using a word processor to do the actual writing. One may conjecture that such an arrangement is probably inefficient: perhaps the scribe is doing much work, but the other co-authors may become distracted, or would prefer to start writing the ideas to order them and propose a consistent composite paper. Of course, this does not imply that any other computer-supported approach will be better in all dimensions, since there are process losses (Nunamaker et al. 1993). For instance, one scheme in which every co-author can write may produce articles with little style coherence. Efficiency benefits may be also found in workflow implementations, since one of the objectives for its adoption within the organization is to improve the efficiency to achieve its goals (Khoshafian and Buckiewicz 1995).

This paper presents a first formal approach to evaluate CSCW applications. This approach does not pretend to be applicable to all CSCW projects or comprehensively evaluate all aspects of them. Nor do we focus on evaluating the CSCW software or the hardware or the software components. Rather, we concentrate on the relationships among issues such as the quality of the outcomes of a CSCW project, the time spent on it, and the total amount of work done.

Evaluation of a CSCW project is, of course, very important from a managerial point of view. In the first place, evaluation before a project is started is a key information to decide whether the project is worthwhile. . Afterwards, evaluation is also useful as a basis for rewarding participants, to justify financing similar projects, or to justify a second phase of the project.

Grudin (1989) has clearly illustrated failed cases of CSCW applications. Benefits that are only for one person are a clear example of failure: why should the other participants be willing to contribute in this case? This is different from the traditional management information system, in which people are not supposed to benefit from the system's operation: their job is to do very specific tasks (for instance, feed the system with certain data), independently of who are the beneficiaries.

Let us assume that the *quality* of a CSCW project could be assessed by some means, such as those summarized by Andriessen (1990). Still, a manager has some very relevant questions to ask, such as, "Could the project have been equally successful with fewer people involved?" Of course, some CSCW projects might have some benefits that are not subject to this type of evaluation; for instance, a joint project may have as its main goal to improve the cohesiveness and sense of community of the members of a department.

An example of the type of projects we would wish to analyse is the following: suppose a group of five persons is assigned the task of doing a computer-aided design project; all members are experts in their fields of specialization and have known each other for some time. For this type of project we would wish to study the relationships among work quality, total amount of work and duration, and number of people involved. Below, we present a detailed example of the applicability of our approach, formalizing what we and other authors call collaborative retrieval. That is, a group of people trying to find at the same time some information needed by the group. There have been some independent attempts on this problem, but it has not been properly formalized regarding retrieval strategies and their performance (Swigger and Hartness 1996, Karamuftuoglu 1998, Sandusky *et al,* 1998). Two of these attempts appeared after a preliminary version of this paper was presented in a conference (Baeza-Yates and Pino 1997).

Basic framework

We model a certain groupware application as a task to be performed by m persons who have the same abilities and task performance. We assume, without loss of generality, that every person does the same amount of work for the given task, so task distribution is also homogeneous. Heterogeneous users and workload distribution can be handled by straightforward extensions to our model. The task is divided into n stages ($n > 1$), which models the intrinsic partitionable nature of most groupware tasks (for example, collaborative writing or design). Our team organization roughly corresponds to the *synchronous* organizational paradigm proposed by

Constantine ([1993](#)).

Stages can be time-based (days), location-based (meetings) or task-based (sub-projects) and are usually well defined. Other general dependencies exist in the real world (for example, modelled by [directed acyclic graphs](#)), and these can be considered as possible extensions of our proposal. In that case, the graph can be divided into layers and linearized to match our simpler model.

Our goal is to measure the efficiency of the task performed by the group. For this, we define the following concepts:

- Quality: how good is the result of the collaborative work? Improving quality was one of the first motivations of CSCW applications ([Ellis et al. 1991](#)).
- Time: the total time elapsed while working. In the real world, the time span is an important factor. In groupware applications, the time to achieve a task can be decreased by doing work in parallel.
- Work: the total amount of work done. When there is no parallelism, the total amount of work done is proportional to the time spent. When there is parallelism, we define work as time spent times the number of persons involved (analogous to work complexity in parallel algorithms ([JaJa 1992](#))).

Project Type	Example	Location	Stages
Sequential	Electronic mail collaboration	Distributed	Days
Parallel and Synchronous	Collaborative writing Negotiation Collaborative design	Both Local Both	Days Meetings Subprojects
Parallel and Asynchronous	Collaborative retrieval	Distributed	Hours

Table 1: Examples of CSCW applications related to our model.

A basic CSCW taxonomy ([DeSantis and Gallupe 1987](#)) relates applications to the location of the users (same or different location) and the time when the collaborative work is carried out (same time or different time). When we have parallelism, the interaction can be synchronous (same time) or asynchronous (different time). We first discuss how this taxonomy fits our framework. Table [1](#) shows some examples of applications, their basic stages and the location of the users.

Location is not strongly related to efficiency. It can be claimed that being in the same location improves quality and may decrease time and work, as the communication among the group members is better than in the distributed case. Nevertheless, this fact will be embedded in the performance measures we use and then the same efficiency analysis applies to CSCW applications independently whether or not they are distributed. On the other hand, using parallelism or a distributed setting not only makes the quantitative analysis more difficult, but the implementation of the application as well.

When parallelism is used, our model does not really depend upon whether or not the work is carried out at the same time. Working synchronously may improve quality, but, again, the analysis remains the same. The coordination points at which every person finishes his or her assignment can constitute the milestones separating stages. On the other hand, if a stage is asynchronous, the group's output for the stage must be integrated and made coherent afterwards. This *unification* activity is the complex part of this collaboration strategy. Further discussion of this topic is presented in the case study below.
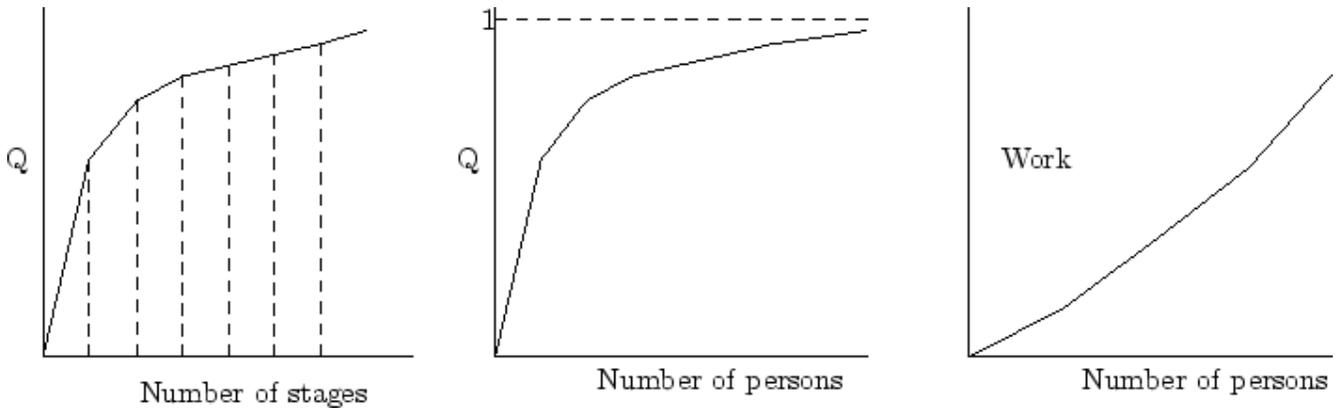
Figure 1: Quality vs. number of stages (left), quality vs. number of persons (middle), and work vs. number of persons (right).

Performance analysis

We first focus on quality. The first important remark, is that for many tasks quality is difficult to assess in general, independently of having or not a collaborative environment. Therefore, our framework only applies to problems where quality can be defined and measured.

As we stated in the previous section, quality improvement analysis is independent of location or parallelism. In most cases, the quality improvement rate decreases with every stage. In the case in which the number of stages is the number of persons involved, there is an optimal number of persons in the sense that the cost of including one more person is not worth the quality improvement. We discuss this issue again, when we look at the time spent in the task. Figure 1 shows an example of prototypical curves.

We want to maximize quality per work done. As usually work increases with the number of people at a rate higher than linear but quality increases at a rate less than linear, there is an optimal number of people which maximizes quality versus work. Considering that usually there is a minimal quality that has to be obtained, we have to choose the maximum of both cases. Figure 2 below (left) illustrates this optimal point, which basically represents where perfectionism starts to be inefficient.

Consider the following simple example of a model for quality $Q=1 - e^{-m/\alpha}$, where 1 is the maximal quality and α measures how fast the collaborative task is saturated by m people. Thus, a local task should have larger α than a distributed task, as communication among the group members is easier. We can maximize Q/m, assuming a simple model where the work is proportional to the number of users. An approximate solution to the optimal number of people to maximize the quality achieved per people ratio is $m \approx 3\alpha/2$
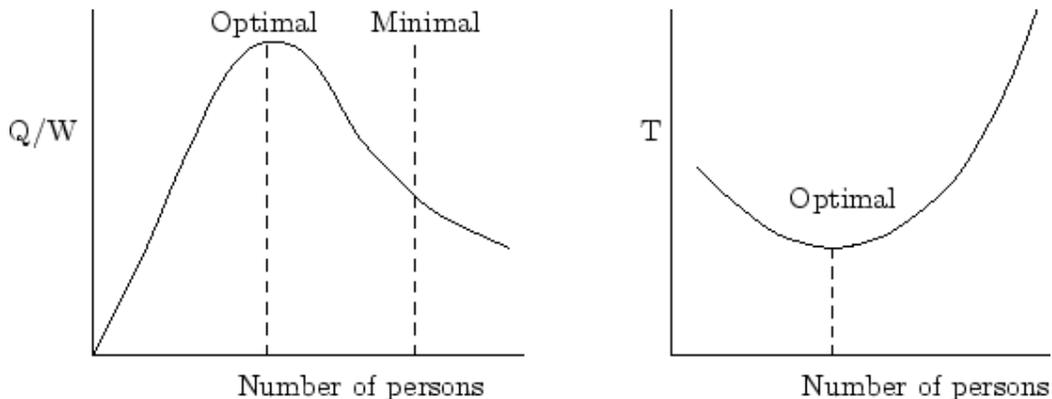
Figure 2: Quality of work vs. number of persons (left) and time vs. number of persons (right).

When no parallelism is used, collaborative work can only improve quality, but will not reduce the time span nor the work done. In sequential work, the task is passed over to the next person until it is finished. One advantage of sequential collaborative work is that no work is repeated. Clearly, if sequentiality is not intrinsic to the application, parallelism should be used if a reduced overall time is a goal.

In the parallel case we can optimize either quality of work or the total time elapsed. We have already discussed the first in the example shown before, where we used the number of persons to represent the total work done. As in many activities, including additional people helps to decrease time. However, this is true up to a certain limit, when saturation implies that adding other personnel actually delays the project's completion deadline. An early example of this is the software development project discussed by Brooks ([1975](#)). The explanation for that apparent paradox is that the coordination effort for increasingly large human teams grows very fast, outpacing any increase in technical work done by the additional personnel. We can do an analogy to parallel algorithms for which the communication load becomes relevant in the overall running time ([JaJa 1992](#)). Note also that the amount of work to be done increases faster than linear because we have to add the interaction time and other losses ([Finholt et al. 1990](#)), such as interruptions ([van Solingen et al. 1998](#)) or work repetition due to synchronization problems. Figure 2 (right) shows an example of time optimality for a task where parallelism saturates due to a superlinear interaction time.

Case study: collaborative retrieval

In this section we apply the ideas outlined above to the specific problem of many people searching together.

Searching strategies

Suppose one would like to do collaborative information retrieval. One approach to this subject is what has been called *collaborative filtering* ([Goldberg et al. 1992](#)): when one person searches, s/he is provided with information that has been useful to other people previously. The corresponding systems have been more adequately re-named as *recommender systems* ([Resnick and Varian 1997](#)).

Our approach does not involve recommendations. We consider a group of people who have been assigned the task of searching for the same information. This case can be viewed as a straightforward extension to the traditional information retrieval mode of only one person searching. There is independent work related to this problem, but its focus is not on searching strategies nor their evaluation ([Swigger and Hartness 1996](#), [Karamuftuoglu 1998](#), [Sandusky et al. 1998](#), [Churchill at al. 1999](#)).

The improvement over single-person search is that several persons doing the job would state the sought items with their own language terms, thereby increasing the *recall* (proportion of all the relevant material retrieved ([Swigger and Hartness 1996](#); [Salton and McGill 1983](#))) of the search process. That is, the knowledge of all the participants acts as a larger thesaurus. This would have practical applicability especially in projects for which a high recall is needed. An example of this need reported in the literature is the case of legal information retrieval discussed by Blair and Maron ([1985](#)). A hypothetical example is a search over the Internet done by several people at the same time, with the help of a visual collaborative tool that keeps track of the global state of the search. It may be noted that in many cases there is no thesaurus easily available and thus collaborative search may make sense as a procedure to increase recall.

An interesting experiment concerning Internet search is reported by Bates ([1998](#)). The 2-term query *+"freedom of speech" +Internet* and three variations of it ("+First Amendment" "+Web", "free speech" "+cyberspace", +"intellectual freedom" "+Net") were run on the AltaVista search engine. The first screen of ten retrievals for each of the queries does not have any intersection with the other three: forty different addresses were obtained in total. Moreover, expanding the search by combining the eight different terms in all the logical combinations gives sixteen new queries. Processing those queries provides 138 unique different entries (of the 160 possible

retrievals). The author concludes: "...thus, if each of the 16 queries had been entered by a different person, each person would have missed 128 other 'top 10' entries on essentially the same topic, not to mention the additional results that could be produced by the dozens of other terminological and search syntax variations possible on this topic" (Bates, 1998: 1189).

After the items have been retrieved in parallel by each of the group participants, there is a unification process where all the sought sets of retrieved documents are merged into one set. Several strategies to achieve that process can be designed. A trivial one, for instance, could be to define the final set as the union of the individual sets of retrieved documents. For this strategy, the *precision* (proportion of the answer that is relevant) of the resulting set will probably be low, which in certain cases is acceptable, provided the recall is high (Salton 1986). Precision and recall are difficult to estimate (Baeza-Yates and Ribeiro-Neto 1999), but there are standard test sets for this task.

Other strategies could involve all participants to refine the final set (even if it is initially defined as the union of the individual sets) in order to improve its precision. Various ways to achieve this are possible, for example, discussion and negotiation of the procedures or language terms which eliminate items from the final set, sequential refinement by the participants, etc.

For our case study, we analyse a particular strategy, the Union-Refine (or U-R) which is stated as follows:

Each user i searches the database individually (in parallel), generating a set of retrieved documents S_i. These sets have recall r_i and precision p_i. Afterwards, an intermediate set T is defined as the r_T union of the sets S_i. Set T has recall r_T and precision p_T.

Through a collaborative step (unification), the users refine set T creating set F by modifying the search statement to eliminate non-relevant documents from T. This can be done by excluding suffixes from the search terms, adding AND terms, etc. Set F gets recall r and precision p.

The described procedure defines one stage on the search task. Since each user works asynchronously, we can think that there is always a *current* set T and users update it whenever they are ready. This simplifies the unification process.

The following can be easily verified:

> P1. $p_T \leq max_i \{p_i\}$
> P2. $p \geq p_T$
> P3. For all i, $r_T \geq r_i$
> P4. $r \leq r_T$
> (almost the same; it is not equal because manipulation of the search terms during unification may affect also the retrieval of relevant documents)

Two interesting measures can be defined to evaluate the collaborative/group quality of the search:

- Group precision: $GP = p/max_i \{p_i\}$
- Group recall: $GR = r/max_i \{r_i\}$

Taking into account properties P1-P4, it is clear that $GP \leq 1$ and $GR \geq 1$. The bigger these measures are, the better the collaboration performance over the individual contributions. Our performance analysis can be applied by using $Q = GP$ or $Q = GR$ as quality measures. For example, we can have a quality per work done defined as $Q / f(m)$, where $f(m)$ gives the financial cost of having m people searching. The optimal number of searchers is obtained maximizing $Q / f(m)$. This can be done analytically if we have a model for Q or experimentally by measuring GP or GR with different numbers of searchers on similar retrieval tasks. In both

cases we assume the searchers have similar abilities and experience.

The analysis can also point out how to improve the unification part, and which heuristic is the best. For example, a quantitative selection based on the number of users approving a subset.

Experiments

A simple set of experiments was made to observe the number of different search terms groups of users may generate in collaborative retrieval. Since this was the only goal, the experiments did not include the succeeding union-refinement of the retrieved documents.

The subjects were college senior students and were given the task of searching a bibliographic database located at RENIB (Centro de Documentacion de Bienes Patrimoniales, Chile) for documents that might be relevant to a certain need. The students were proficient in Spanish and thus they were asked to state search terms in that language (the database also contains documents in Spanish).

Table 2 shows results for individual searches. Ten subjects were partitioned in three 'groups'. The participants of each group did not have communication among them and had to find documents which they would assess as relevant for the problem at hand, based on the title and abstract of the retrieved bibliographic records. The students were asked to turn in the search terms they used and the ones found really useful in finding the relevant documents.

Measure	Group I	Group II	Group III
Number of participants	3	4	3
Number of searches performed	11, 9, 7	19, 9, 13, 40	10, 19, 13
Avg. No. of searches	8.1	20.3	14
Total number of search terms	19, 7, 5	36, 14, 13, 34	15, 22, 16
Number of search terms found useful	13, 5, 5	20, 12, 11, 25	8, 8, 10
Avg. number of useful search terms	7.7	17	8.7
Relevant docs. retrieved	24, 7, 6	11, 8, 12, 24	51, 28, 61
Avg. number of relevant docs.	12.3	13.8	46.7

Table 2: Experimental results (numbers separated by commas refer to different participants.)

As may be seen from Table 2, the work among groups varies (probably reflecting differences in task complexity) and within each group (reflecting individual abilities and labour). The document relevance was probably judged differently by each subject and thus we obtain very different numbers of relevant documents retrieved.

Nineteen other subjects were partitioned in five groups. These were real work groups, in the sense they had a joint task. They had some type of parallel work but had to coordinate themselves and turn in a group result.

Groups A and B worked on different problems, but had the same work organization: one person worked alone and passed the results to two other participants who worked in parallel; finally, a fourth participant received the results of the two previous students and produced the final result.

Groups C and D worked again on different problems but had this same organization: three persons worked in parallel but communicated with a coordinator who suggested search terms successfully tried by the members of the group. The coordinator also assembled the final results. Group D had the same problem to work on as

group B.

Finally, Group E had three participants working in parallel communicating among themselves with no coordinator. The results are shown in Table 3.

Group	A	B	C	D	E
Number of searches performed	14	24	8	9	10
Total number of search terms used	32	19	16	8	10
Number of search terms found useful	17	14	14	8	9
Relevant documents retrieved	10	53	32	50	80

Table 3: Relevant documents retrieved.

Group A worked on the same problem as Group II, Group C on the same problem as Group I and Group E on the same problem as Group III. As can be observed, in this experiment at least, the joint work had more relevant documents retrieved than the corresponding individuals (of groups I, II and III) working alone. Of course, this cannot be assured since the students themselves assessed relevance. It is interesting to note that although the number of useful search terms is similar between groups working in the same problem, the number of relevant retrieved documents is higher in the case of collaborative groups. In particular, with respect to the non-collaborative experiments, groups C and E improved significantly the performance by using collaboration, which suggests that communication either with or without coordination makes a difference.

Discussion and conclusions

Our work has concentrated on a very limited collaboration scenario. This is because the group members are supposed to make similar contributions to the joint work. This is not frequently found in practice, since most collaborative works benefit from the variety of contributions from the group members. Different members' backgrounds, roles, attitudes and interests contribute to the richness of the joint work. "Nobody's perfect, but a team can be" says Belbin's well-known method building a team with several role specializations (Belbin 2003). An example of rich team building with various interests and backgrounds is provided in the AMI@Work initiative.

The collaboration scenario presented in the previous sections, instead, may be applicable to cases where work can be done interchangeably by people. The collaborative information retrieval case is an example of that. Another instance may be the work carried out by employees of a call centre answering requests from valuable customers, where the same type of work done in parallel for one customer could be justified.

The collaborative information retrieval example shows how it is possible to apply a formal evaluation, in the sense defined here, and analysis to a groupware application. This type of analysis may imply savings in total time span, number of people involved in a project, etc. and may complement qualitative evaluations made over the project. Evaluations of this type would then be useful to managers.

The example showed a simple solution to the unification problem. In other cases, the unification problem could become complicated, for instance, when a variable number of persons interact to consolidate the outcome of one stage. For these cases, the type of analysis we have presented will probably be unsuitable.

The applicability of this type of formal analysis is also limited by the availability of quantitative data concerning the application. For instance, we need to know individual task performances: if we do not know anything about this beforehand it is impossible to do any computation.

The *structure* we assumed for the CSCW applications is relatively small compared to workflow applications, in

which the flow of information is optimized in order to improve the results of a business process (Burns 1994). Further work will apply our model to existing collaborative applications as well as lifting some of its limitations.

Acknowledgements

References

- Andriessen, J.H.E. (1990). The why, how and what to evaluate of interaction technology: a review and proposed integration.In J. Galegher, R.E. Kraut, and C. Egido, (Eds.). *Intellectual teamwork*, (pp. 107-124). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Belbin, M. (2003). *Management teams: why they succeed or fail*. 2nd ed. London: Butterworth Heinemann.
- Bates, M. (1998). Indexing and access for digital libraries and the Internet: human, database, and domain factors.*Journal of the American Society for Information Science* , **49**(13), 1185-1205.
- Blair, D.C. & Maron, M.E. (1985). An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM*, **28**(3), 289-299.
- Brooks, F.P. (1975). *The mythical man-month*. Reading, MA: Addison-Wesley.
- N. Burns, N. (1994). *Workflow automation: a methodology for change. The workflow CD-ROM sampler*. Palo Alto, CA: Creative Networks.
- Baeza-Yates, R. & Pino, J.A. (1997). A first step to formally evaluate collaborative work. In *Proceedings of the international ACM SIGGROUP conference on supporting group work: the integration challenge, Phoenix, Arizona, USA.*, (pp. 56-60). New York, NY: Association for Computing Machinery.
- Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern information retrieval*. Harlow, UK: Addison-Wesley-Longman.
- Churchill, E., Sullivan, J., Golovchinsky, G. & Snowdon, D. (1999). Collaborative and co-operative information seeking: CSCW'98 Workshop Report, *SIGGROUP Bulletin*, **20**(1), 56-59.
- Constantine, L.L. (1993). Work organization: paradigms for project management and organization. *Communications of the ACM*, **36**(10),35-43.
- DeSantis, G. & Gallupe, G. (1987). A foundation for the study of group decision support systems. *Management Science*, **33**(5), 589-609.
- Ellis, C.A., Gibbs, S.J. & Rein, G.L. (1991). Groupware: some ideas and experiences. *Communications of the ACM*, **34**(1), 38-58.
- Eason, K. & Olphert, W. (1996). Early evaluation of the organisational implications of CSCW systems. In P. Thomas, editor, *CSCW requirements and evaluation*, (pp. 75-89). London: Springer.
- Finholt, T., Sproull, L. & Kiesler, S. (1990). Communication and performance in ad hoc task groups. In Galegher, J., Kraut, R.E. & Egido, C. (Eds.). *Intellectual teamwork*, (pp. 291-325). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Goldberg, D., Nichols, D., Oki, B. & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, **35**(12), 61-70.
- Grudin, J. (1989). Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. *Office: Technology and People*, **4**(3), 245-264.
- JaJa, J. (1992). *An introduction to parallel algorithms*. Reading, MA: Addison Wesley.
- Karamuftuoglu, M. (1998). Collaborative information retrieval: towards a social informatics view of IR interaction. *Journal of the American Society for Information Sciences*, **49**(12), 1070-1078.
- Khoshafian, S. & Buckiewicz, M. (1995). *Introduction to groupware, workflow, and workgroup computing*. New York, NY: John Wiley and Sons.
- Nunamaker, J.F., Dennis, A., Valacich, J.S., Vogel, D.R. & George, J.F. (1993). Group support systems research: experience from the lab and field. In Jessup, L.M. and Valacich, J.S. (Eds.). *Group support

systems - new perspectives, (pp. 125-145). New York, NY: Macmillan.

- Resnick, P. & Varian, H.R. (1997). Recommender systems. *Communications of the ACM*, **40**(3), 56-58.
- Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM*, **29**(7), 648-656.
- Swigger, K.M. & Hartness, K. (1996). Cooperation and online searching via a computer-supported cooperative problem solving environment. *Journal of the American Society for Information Sciences*, **47**(5), 370-379.
- Salton, G. & McGill, M.J. (1983). *Introduction to modern information retrieval*. Singapore: McGraw-Hill.
- Sandusky, R., Powell, K and Feng, A. (1998). Design for collaboration in networked information retrieval. In *Proc. of the ASIS Mid-year Meeting*, (pp. 95-105). Orlando, FL, USA.
- van Solingen, R., Berghout, E. & van Latum, F. (1998). Interrupts: just a minute never is. *IEEE Software*, **15**(5), 97-103.