A First Step to Formally Evaluate Collaborative Work

Ricardo Baeza-Yates José A. Pino
Depto. de Ciencias de la Computación
Universidad de Chile
Casilla 2777, Santiago, Chile
E-mail:{rbaeza,jpino}@dcc.uchile.cl

Abstract – *We present an initial attempt to formally evaluate performance measures related to CSCW applications. In particular, we study the relations between the quality of the output, the number of people and time spent on the overall task and the total work done. We introduce a novel application to illustrate our ideas: collaborative retrieval. We propose a specific heuristic to the case when many people search for the same information, thereby increasing the recall and precision of the answer. We apply our evaluation methodology to this case.*
Keywords: CSCW, quantitative evaluation, collaborative retrieval.

Introduction

The introduction of Computer Supported Cooperative Work (CSCW) tools within an organization can have a variety of positive outcomes [1]. Generally speaking, those benefits can be classified into three broad categories.

The first type of benefits of CSCW introduction refers to an improved quality of the end product. Better documents, designs or decisions may be the result of the use of collaborative editors, CAD or electronic meeting systems, respectively, when compared with similar outcomes obtained with computer applications intended for individual use.

A second kind of CSCW benefits concerns 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.*[2] mention the satisfaction of the group members with the result, which was a composite of individual contributions.

The third type of gains has to do with efficiency: to reduce wasted time or opportunities for people to effectively contribute to a joint product. Consider, for example, 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 and just one of them using a word processor to do the actual writing. One may conjecture that such an arrangement probably is inefficient: perhaps the scribe is doing much work, but the other co-authors may get distracted, or would prefer to start writing the ideas to order them and propose a consistent aggregate. Of course, this does not imply that any other computer-supported approach will be better in all dimensions, since there are process losses [3]. 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 its efficiency to achieve its goals [4].

This work was supported by Fondecyt (Chile) Grant No. 197-0342.

This paper attempts to present a formal approach to evaluate Computer-Supported Cooperative Work (CSCW) applications. This approach does not pretend to be applicable to all CSCW projects or comprehensively evaluate all aspects of them.

We do not focus on evaluating the CSCW software or the hardware or software components of it either. We rather concentrate on the relationships among issues like 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 worth to be carried on. Afterwards, evaluation is also useful to reward participants, to finance similar projects or to continue the project in a second phase.

Grudin [5] has clearly illustrated failed cases of CSCW applications. Benefits which are only for one person are a clear example of that: why should the other participants be willing to contribute in this case? This is different from the traditional Management Information Systems, 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 [6]. Still, a manager has some very relevant questions to ask, such as "could the project had been equally successful with less personnel involved?". Of course, some CSCW

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 I

Examples of CSCW Applications related to Our Model.

projects might have some benefits which 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 group of the members of a department.

An example of the type of projects we would like to analyze is the following: suppose a group of five persons is assigned the task of doing a CAD project; all members are experts in their fields of specialization and know each other for quite some time. For this type of projects we would like to study the relationships among work quality, total amount of work and duration, and number of people involved. In Section 4 we present a detailed example of the applicability of our approach, introducing a new problem which we call collaborative retrieval. That is, a group of people trying to find at the same time some information needed by the group.

Basic Framework

We model a certain CSCW 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 CSCW tasks (for example, collaborative writing or design). Our team organization roughly corresponds to the *synchronous* organizational paradigm proposed by Constantine [7].

Stages can be temporal based (days), location based (meetings) or task based (subprojects); and are usually well defined. Other general dependencies exist in the real world (for example, modeled 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.

To measure the efficiency of the application, 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 [2].
- Time: the total time elapsed while working. In the real world, the time span is an important factor. In CSCW 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 [8]).

A basic CSCW taxonomy [2] 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 I 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 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 if they are distributed or not. On the other hand, using parallelism 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 if the work is carried out at the same time or not. Working synchronously may improve quality, but again, the analysis remains the same. The coordination points at which every person finishes his/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 Section 4.

Performance Analysis

We first focus on quality. As we stated in the previous section, quality improvement analysis is independent of location or parallelism. In most cases, 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 opti-

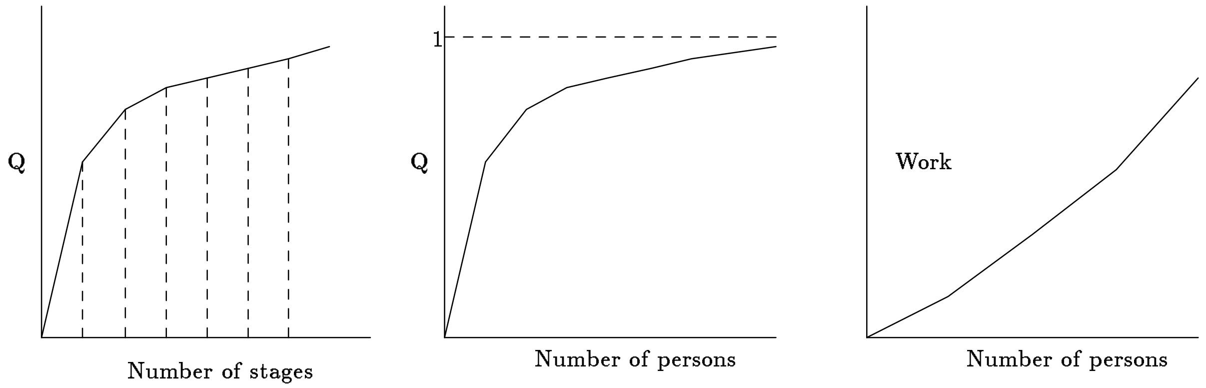

Fig. 1. Quality vs. number of stages (left), quality vs. number of persons (middle), and work vs. number of persons (right).

mal 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 work increases with 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 (left) illustrates this optimal point.

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 have to maximize Q/m, because the work is proportional to the number of users. An approximate solution to the optimal number of people is $m \approx 3\alpha/2$.

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 per work or time. We have already discussed the first. As in many activities, including additional people helps to decrease time; 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 [9]. 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 [8]. 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 [10]. Figure 2 (right) shows an example of time optimality.

Case Study: Collaborative Retrieval

Suppose one would like to do collaborative Information Retrieval. One approach to this subject is what has been called Collaborative Filtering [11]: when one person searches, he/she is provided with information which has been useful to other people previously. The corresponding systems have been more adequately re-named as "Recommender Systems" [12].

Our approach does not involve recommendations. We consider a group of people who have been assigned the task of searching the same information. This case can be viewed as a straightforward extension to the traditional Information Retrieval of only one person searching. 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 relevant material retrieved [13]) 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 [14]. Another 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.

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 retrieved material that is relevant) of the resulting set will probably be low, which in certain cases is acceptable, provided the recall is high [15]. Precision it is difficult to estimate [15], but there are standard test sets for this task.

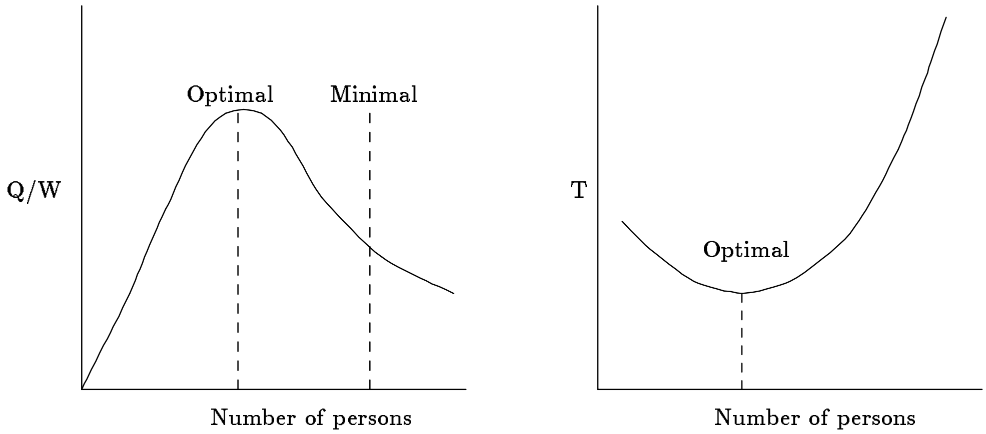

Fig. 2. Quality per work vs. number of persons (left) and time vs. number of persons (right).

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 analize 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 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 in order 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 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 number of searchers on similar retrieval tasks. In both cases we assume that the searchers are homogeneous.

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. These ideas will be explored in a subsequent paper.

Concluding Remarks

The collaborative Information Retrieval example shows how it is possible to apply a formal evaluation - in the sense we defined - and analysis to a CSCW 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 get 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 [16].

Further work will apply our model to existing collaborative applications as well as lifting some of the limitations of it.

REFERENCES

[1] K. Eason and W. Olphert. Early evaluation of the organisational implications of cscw systems. In P. Thomas, editor, *CSCW Requirements and Evaluation*, pages 75–89, London, 1996. Springer Verlag.

[2] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware: Some ideas and experiences. *Comm. of the ACM*, 34(1):38–58, 1991.

[3] J.F. Nunamaker, A. Dennis, J.S. Valacich, D.R. Vogel, and J.F. George. Group support systems research: Experience from the lab and field. In L.M. Jessup and J.S. Valacich, editors, *Group Support Systems - New Perspectives*, pages 125–145, New York, NY, 1993. Macmillan Publishing Co.

[4] S. Khoshafian and M. Buckiewicz. *Introduction to Groupware, Workflow, and Workgroup Computing.* John Wiley and Sons, New York, NY, 1995.

[5] J. Grudin. Why cscw applications fail: Problems in the design and evaluation of organizational interfaces. *Office: Technology and People*, 4(3):245–264, 1989.

[6] J.H.E. Andriessen. The why, how and what to evaluate of interaction technology: a review and proposed integration. In J. Galegher, R.E. Kraut, and C. Egido, editors, *Intellectual Teamwork*, pages 107–124, Hillsdale, NJ, 1990. Lawrence Erlbaum Associates, Inc.

[7] L.L. Constantine. Work organization: Paradigms for project management and organization. *Communications of the ACM*, 36(10):35–43, Oct 1993.

[8] J. JaJa. *An Introduction to Parallel Algorithms.* Addison Wesley, Reading, Mass., 1992.

[9] F.P. Brooks. *The Mythical Man-Month.* Addison-Wesley, Reading, MA, 1975.

[10] T. Finholt, L. Sproull, and S. Kiesler. Communication and performance in ad hoc task groups. In J. Galegher, R.E. Kraut, and C. Egido, editors, *Intellectual Teamwork*, pages 291–325, Hillsdale, NJ, 1990. Lawrence Erlbaum Associates, Inc.

[11] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[12] P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[13] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Singapore, 1983.

[14] D.C. Blair and M.E. Maron. An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.

[15] G. Salton. Another look at automatic text-retrieval systems. *Communications of the ACM*, 29(7):648–656, 1986.

[16] N. Burns. *Workflow automation: a methodology for change.* The workflow CD-ROM sampler. Creative Networks, 1994.