Adoption Patterns of Low-structure Groupware: Experiences with Collaborative Writing Software

V. Srinivasan Rao
Division of Accounting and Information Systems
University of Texas at San Antonio
San Antonio, TX 78249.

Poppy L. McLeod
Department of Management and Organizations
The University of Iowa
Iowa City, IA 52242

Kathy M. Beard
School of Business Administration
University of Michigan
Ann Arbor, MI 48109.

Abstract

Longitudinal use of a collaborative writing software was observed to identify patterns of adoption. Use of the software was limited and patterns of adoption inconsistent with expectations. In this paper, we propose possible reasons for the observed patterns. The possible reasons fall within the broader arguments of cost-benefit analyses and task-technology fits. However, our analysis suggests that violations of the cost benefit and task technology fit criteria may be occurring at multiple levels, e.g., at the global task level and the sub-task level. We argue that effective and appropriate adoption of technology requires the cost-benefit balance and the task-technology fit to be favorable at all levels. Recommendations for improvements in design are made.

Introduction

The assimilation of new group technologies into existing group structures is a process of action and interaction that leads to new emergent patterns of group structure (Poole, 1985). DeSanctis and Poole (1994) state 'people actively select how technology structures are used, and adoption practices vary' (p. 129). They enumerate four choices that the groups may make in the appropriation of the new technology structures, --- directly use the structures, relate the structures to existing task or environmental structures, constrain or interpret structures as they are used, and make evaluative judgments about the structures. No explicit reference is made to the possibility of non-use, but it would be fair to assume that that possibility exists. In this paper, we (a) report our observations and interpretations of the adoption patterns of collaborative writing software, (b) make recommendations for the design of collaborative software and their study.

Patterns of the adoption of group support systems (GSS) have been described by Zigurs, DeSanctis and Billingsley (1990). Over a span of eight meetings, in which groups performed two different tasks, Zigurs et al observed that the extent of usage dropped by more than 60% for the adopter groups and more than 80% for the discarder groups. The reduction in usage over time is surprising for two reasons. First, several studies of groups using GSS for single sessions have reported productivity gains (Lewis, 1982; Gallupe, DeSanctis and Dickson, 1988; Nunamaker et al, 1989). Second, it would have been logical to assume that over time the groups would have become more familiar and comfortable with the technology, and consequently used it more.

One possible reason for the limited use of the GSS is put forth by McLeod and Liker (1992). They argue that GSS is a high-structure system, i.e., systems designed specifically to change a group's preexisting structures...these systems offer specific methods for communication and approaching the task' (p. 197). High-structure systems are relatively inflexible. Hence groups find it difficult to incorporate the new technology structures offered by these systems into existing group structures. Consequently, groups tend to ignore the new structures or use them to a limited extent.

Alternately, low-structure systems are 'not designed to change group structure in any specific way...the degree to which a change in...the pattern of interacting...occurs depends on groups creating structures for themselves, using tools provided by the system.' (McLeod and Liker, 1992, p. 197). Low-structure systems are relatively flexible. It can be argued that groups would find it easier to incorporate the technology structures offered by these systems into existing group structures. Consequently, a high level of adoption can be expected.

We chose ShrEdit (Killey, 1990; McGuffin and Olson, 1992), a collaborative writing software as a representative low-structure system to explore this expectation. ShrEdit is not built on any strong model of collaborative processes (Dourish and Bellotti, 1992) and allows users to adapt the tool to their particular style of working (McGuffin and Olson, 1992). There is evidence that collaborative writing tools are effective when evaluated in single-sessions lasting one to three hours (Olson, Olson, Storrssten and Carter, 1992a; Dourish and Bellotti, 1992), but there is no longitudinal information on the adoption of these tools.

We emphasize that our intent in this paper is to explore, i.e., provide possible explanations for what was observed. The explanations do not constitute definitive findings, but can be used for building theories and developing hypotheses, which can then be tested rigorously.
So, in this paper, we report on the adoption pattern of ShrEdit, a collaborative writing tool, over an average of eight sessions of use by groups required to write a critical evaluation of a business case. In section 2, we review some of the pertinent literature and concepts. In section 3, we provide the details of methodology. In section 4, we describe the collaborative processes observed. In section 5, we offer our interpretations of the observed collaborative processes. In section 6, we make recommendations for improvements in the design of existing group support software, and their study.

Literature review

In this section, we review some of the literature on high- and low-structure groupware, empirical results from GSS studies, collaborative writing, and writing phases.

Low-structure, high-structure groupware

The concept of high-structure and low-structure groupware was put forth by McLeod and Likert (1992) in an effort to understand some of the differences reported in literature on the effects of diverse groupware. The key distinction between low- and high-structure systems is the preponderance of influence exerted by groups and technology over each other. A high-structure system is one in which the technological structures are relatively inflexible and existing group structures will have to change considerably if the technology is to be adopted in its true spirit. The agenda in SAMM (Watson et al, 1988; Zigurs et al, 1988) and the one-module-at-a-time process of GroupSystems (Dennis et al, 1988) are examples of high structure. A low-structure system, on the other hand, is one in which the structures provided by the technology are relatively flexible and can be easily accommodated into existing group structures. Examples of low structure systems are Capture Lab, an electronic meeting system (Mantei, 1988; Elwart-Keys et al, 1990; Horton et al, 1991), and collaborative software, such as ShrEdit (McGuffin and Olson, 1992).

An example of differences reported in literature is one related to the relative use of technology by users during a groupware session. Mantei (1989), in a case study of groups working in the Capture Lab (a low-structure system), found that groups’ distribution of technology use was quite unequal. She observed that the pattern of control was somewhat related to social status hierarchies that had existed in the groups before their Capture Lab session. Austin et al (in press), using experimental data, also found distribution of technology control to be quite unequal in groups meeting in the Capture Lab. Their analysis also revealed that technology use and control could be predicted by preexisting social status hierarchy. In contrast, studies of high-structure systems, i.e., group support systems, show that control over technology tends to be equally distributed among members (Dennis et al, 1988; Pinsonneault & Kraemer, 1989; McLeod, 1992). The two sets of studies collectively indicate that the existing hierarchical structures in the group are better accommodated by the low-structure groupware than by the high-structure groupware.

Empirical results from studies on GSS

There are two aspects of empirical studies of group support systems that are relevant: evidence of the effectiveness of group support systems in single-session studies, and the adoption of group support systems in longitudinal studies.

Literature is replete with conflicting evidence of the effectiveness of GSS in single-session studies. Some studies have found positive results (e.g., Nunamaker et al, 1989; Lewis, 1982; Gallupe, DeSanctis and Dickson, 1988), while others found neutral or negative results (Watson, DeSanctis and Poole, 1988; Turoff and Hiltz, 1982; Jarvenpaa, Rao and Huber, 1988). However, meta-analyses of these studies have provided support for the argument that GSS are effective under certain conditions (see Benbasat and Lim, 1993; McLeod, 1992).

Evidence of adoption patterns comes from longitudinal observations of students performing term projects in an university environment (Zigurs et al, 1990) and observations of ongoing groups in the organizational environment (for example, Poole, DeSanctis, Kirsch and Jackson, 1991; Vician, DeSanctis and Poole, 1992). Some observations are common to both studies. First, there is considerable variation in the behavior across groups within each study. For instance, Zigurs et al were able to divide the eight groups in their study into adopters and discarders based on the extent to which the groups use the available technology. Similarly, Poole et al observed that some groups learnt and appropriated all the structures available in the GSS, while other groups did not use any of the structures. Second, even the adopter groups used the systems only to a limited extent. In the Zigurs et al study, usage declined considerably over the eight meetings. In the Poole et al study, usage fluctuated over the series of meetings.

Collaborative Writing

The design of most collaborative writing software has been based on formal or informal analysis of groups collaborating on a writing task without any technological support (Olson et al, 1992a; Posner, Baeker and Mantei, 1991). For instance, Olson et al (1992a) determined that the difficulties observed in the unsupported environment could be alleviated by having a shared, editable display of the group’s current working object. Since collaboration was perceived as a free-flowing task, the designers did not generally embed any particular model of collaborative processes or require a facilitator. Thus ShrEdit was designed to be simple and flexible, which allowed users to adapt the tool to their particular style of working (McGuffin and Olson, 1992).
Writing Phases

There is a large literature on the process of collaborative writing (Ede & Luusford, 1990; Journal of Business Communication, special issue, 1987). This literature has focused attention on such diverse topics as cognitive processes in writing (Hayes & Flower, 1980; Flower & Hayes, 1980), interpersonal dynamics involved (Forman & Katsky, 1986) and the stages or phases of the writing process (Hayes & Flower, 1980; Henze, 1984; Adams & Thornton, 1986).

Of relevance to the current study is an understanding of the phases in creating a document. The process of composition proceeds recursively through a series of overlapping stages (Hayes & Flower, 1980; Flower & Hayes, 1980; Forman & Katsky, 1986; Horton et al, 1991). Three basic processes are commonly identified - planning, drafting and revising -- but, there are wide variations in how writers move through them. For instance, writers may combine the drafting and planning stages or break the basic phases into numerous sub-phases (Henze, 1984).

The effect of technology on the progress of groups through these phases has also been studied to some extent. Horton et al (1991), using groups of MBA students writing business memoranda, reported that electronically supported groups spent significantly less time in the planning phase than did non-supported groups. Galegher and Kraut (1990), also using groups of students, found that face to face unsupported groups spent greater effort in initial planning than did remote, computer-supported groups. All these studies suggest that electronic support affects patterns of movement through the phases.

However, in all these studies, the subjects were required to use the technology. In the real world, users have a choice of whether they use the technology. The adoption behavior of subjects when they have a choice has yet to be explored and constitutes the primary thrust of this study.

Conditions of Observation

Three groups were observed while performing a collaborative writing project, which extended over three weeks. Two of the groups were provided access to collaborative software, while the third group was provided access only to conventional, single-user software. Details of the study are provided below.

Superficially, the study has the structure of an experiment, i.e., two groups using the collaborative software and one group not using it. However, the small sample size is inadequate to provide statistically valid results. Our intent is to explore the behavior of the groups over an extended period in search of patterns. The structure used is convenient for comparative observations. Observed differences between the groups using the software and those not using the software have the potential for being interpreted. Admittedly, alternate structures could have been used to observe.

Participants

The participants were volunteers from first-year MBA students enrolled in an Organizational Behavior class at the University of Michigan. As part of the course, students were required to work in groups on two major projects. Each project required the groups to produce a written report. The second of the two projects formed the basis of the study. Twelve groups volunteered to participate in the study. Three of these groups were chosen on the basis of group member experience with computers in general. However, none of the group members had used the collaborative writing tool. Two of these groups were permitted to use ShrEdit, the collaborative writing tool, to produce the report, while the third group was not permitted access to ShrEdit.

Collaborative Software

The collaborative software used was ShrEdit, a shared-editing software package developed at the University of Michigan (Killey, 1990). ShrEdit allows all meeting participants to simultaneously edit a common document. The software prevents two or more persons from editing at the same "insertion point", but does not otherwise restrict the ability of an individual to alter any part of the document. Editing actions are limited to simple commands, such as cut and paste. Selective character formatting is not possible i.e., any formatting (e.g., font choice) must be applied to the entire document. ShrEdit has no graphics capability. The software has a background logging function, invisible to the user.

The Facilities

Collaboration Technology Suite (CTS) is an electronic meeting facility at the University of Michigan. The CTS contains moveable electronic workstations called ELMERS (EElectronic MEeting Stations -- see Cornell et al, 1990). The ELMERS were configured into a rectangular conference table for this study. Each ELMER houses a 17" Macintosh computer monitor in a glass-fronted case. The display case can be adjusted from 0 to 90 degrees from the horizontal, allowing users to adjust their viewing angle. A keyboard and mouse are on the work surface in front of the monitor housing.

The CTS facilities included extensive white board space on two walls, and a rear-projection public display monitor on a third wall. A one-way mirror on the fourth wall joins the CTS to a large observation room.

Training

Groups who were permitted to use the collaborative software (these groups will henceforth referred to as COLL-1 and COLL-2) attended a two-hour training meeting before beginning work on their project. During training, they were
taught how to use ShrEdit and other support software, and the CTS hardware (i.e. how to adjust ELMER screens, how to use the public display screen, how to operate the printer, etc.). None of the participants had any prior experience with ShrEdit. In addition, they were given some general advice on managing the technology. For example, since ShrEdit’s formatting capabilities were limited, the students were advised to use other software, also available in the CTS, for their final formatting (e.g. Microsoft Word©).

The group that was not allowed to use the collaborative software (henceforth referred to as CONV-1) received a brief introduction to the CTS at the beginning of their first meeting but were not given training on any specific software. The participants were already familiar with the wordprocessing package that were necessary to produce the report. We were interested in comparing the process of collaboration with the collaborative software to the typical collaboration of student groups. Typical student collaboration does not occur in facilities such as CTS. However, did need to be able to observe the collaboration process. So we tried to create conditions in the CTS as close as possible to that students may otherwise have. This was done by providing the CONV-1 group with tools that would normally have been available to the students in the school’s computer lab and group study rooms.

Task

The groups were presented with a case study of a Fortune 500 company, written specifically for the assignment. Their task was to analyze the case and to write a memo to the company’s CEO detailing the results of their analysis and making recommendations for actions. The assignment required no outside research by the groups. The groups were allowed three weeks to complete the project. In addition to the grade, there was an additional competitive element. The top twelve papers (out of about 70 from the 12 sections of the MBA class) were actually sent to the CEO who had agreed to read them, and to comment on them during a special program on campus.

Conditions of observation

Three groups were observed. Two of the groups (COLL-1 and COLL-2) were provided ShrEdit, the collaborative writing tool to complete the project. While the third group (CONV-1) was provided conventional single-user word processing software. The third group using conventional word processing software was observed to obtain a representation of baseline behavior of groups working on the assignment.

The groups, COLL-1 and COLL-2, were given full access to all the software and hardware available in the CTS. Group, CONV-1, also met in this room, but was restricted to using tools that they would “normally” have elsewhere in the school. Thus, they were not given access to the shared editing software or public display monitor, but could use one of the room’s computers. They were given full access to the white boards, the printer, and a nearby copy machine.

In both conditions, at least two experimenters were available in the observation room at all times during group meetings. They provided assistance only of a technical nature pertaining to the hardware and software in the CTS, and only when requested or when it was readily apparent that a technical problem was inhibiting a group. They offered no advice on effective uses of the tools.

Data Collection

All meetings were videotaped. The experimenters noted the technology used and any unusual occurrences. They also copied what was written on the white boards. Copies of ShrEdit log files for COLL-1 and COLL-2 groups, and all intermediate and final documents for all groups were saved on disk. Subjects completed a questionnaire, assessing their general background, their attitudes toward and experience with computers and group work, prior to the study. Their resumes were also collected. Following each meeting, they filled out a questionnaire regarding that day’s work. At the end of the study, they completed another questionnaire assessing their opinions of their experience, and participated in group interviews.

This paper relies primarily on experimenter notes, the groups’ intermediate documents, and the post-experiment interview. We have examined these data qualitatively to get an overall picture of the strategies used by these groups to approach their task and support tools, and how the group members interacted.

A comparison of the collaborative processes

The processes of collaboration employed by the three groups are compared to each other in this section. The processes show some similarities but also show some differences. The differences may be a result of the structures provided by the collaborative writing software, or of group member characteristics or other differences between groups. Since our interest is in technology-related issues, we will explore explanations for observed differences between groups using different technologies.

All three groups followed a similar sequence of phases: prewriting, writing/drafting, merging/integrating and completion. These phases are comparable to phases described by others (Hayes and Flower. 1980; Forman and Katsky. 1986). The groups progressed from one phase to the next in almost linear fashion. However, within each phase, there were multiple activities. The activities within each phase do not follow a linear sequence but are generally performed iteratively. There is little or no iteration between the phases themselves.
The prewriting phase

The groups tended to start with generating ideas, and organizing them in some fashion. They proposed sections of the outline. Some composition was done, but the composition was more in the nature of elaborating the ideas generated. Then the groups evaluated what they had done. The evaluations tended to trigger a return to idea generation -- creating more ideas, or adding refinements to existing ones. The prewriting phase was an iterative process in which the groups looped through several activities: generating ideas, planning the document, composing and evaluating.

One of the groups (COLL-1) spent all but two of the meetings on this loop, while the other two groups (COLL-2 and CONV-1) spent over half of their meetings in the prewriting phase.

Both COLL-1 and COLL-2 groups attempted to use ShrEdit in the multi-user mode (i.e., collaboratively) to brainstorm at first, but quickly became overwhelmed by the volume of ideas, and confused and frustrated with so many people working at once in the same free-form document. These groups abandoned the multi-user input strategy after their first meeting, and generally had one group member act as the scribe throughout most of their remaining meetings.

Both COLL-1 and COLL-2 groups abandoned the use of the public display monitor after their first meeting, reporting in their questionnaires that they found it distracting. One of the groups did not turn on the screen during subsequent meetings. The other group continued to turn it on occasionally, but then did not look at it during their meetings. Ironically, during one of their last editing sessions, a member of one of these groups remarked in his post-meeting questionnaire, "It would be great if we could all view the same thing."

The CONV-1 group adopted a scribe strategy right away. One member, using the computer available to them, recorded all their meeting notes. In addition, other members wrote ideas on the white boards, which the scribe usually copied on the computer. However, the scribe did not capture graphic material.

The drafting / writing phase

At the end of the prewriting phase each of the groups had a working outline. The outline was in sufficient detail to permit the groups to segment the paper into discrete chunks for assigning writing responsibilities. The allocation of writing responsibilities was done in two different ways. In COLL-1 and CONV-1, each group member was assigned a segment to write. In COLL-2, one group member was assigned the task of writing the complete draft document. In none of the cases did the group members convene during the writing phase. All work in this phase was done outside the CTS.

The merge / integrate phase

This phase began upon individuals completing their assigned writing responsibilities. This phase included merge, evaluate, compose and adjust activities. The sequence in which these activities were performed tended to vary. In group COLL-1, each group member had a segment of the writing to do. Each segment was first evaluated, edited by a sub-group. When the segment or a part of the segment was considered acceptable, the sub-group would hand over that segment to the person responsible for the keeping the whole document. The subgroups which were editing used ShrEdit, but not in a collaborative manner. Instead, they used it as a single-user text-editing tool. The working pattern of the subgroups alternated between individual and joint work, often with two or more people gathered at the same terminal. The individual who was integrating the document used conventional software, i.e., Microsoft Word®. Group members would gather around his terminal to observe this person and make oral suggestions. In essence the sequence of activities in this phase in this group was evaluate-compose-adjust followed by merge followed by evaluate-compose-adjust.

The group that had assigned one member to write the paper (COLL-2) had no merging activity to perform. They only had to evaluate, edit and compose, as necessary. The physical task of editing was assigned to two members who worked jointly in the lab. They were the only ones who made changes to the paper throughout this loop although other group members checked on them occasionally. This group abandoned ShrEdit at the beginning of this phase and worked exclusively with a standard word processing package.

In group CONV-1, the drafts of all segments were merged first using a conventional word processing package. Subsequently, the group was involved in evaluate-compose-adjust activities. The editing was done by one person at a time, with other group members looking over his/her shoulders to make suggestions.

Overall, the activities in the merge / integrate phase can be divided into two clusters: merge and evaluate-compose-adjust. The merging activity is necessary when the writing is done in segments by different persons, but is otherwise not required. The sequence in which the merge and evaluate-compose-adjust clusters was undertaken varied in different groups. Once groups entered the merge / integrate phase, no further work outside of the group was done, i.e., no individual writing was done and there was little evidence that they returned to the first phase, i.e., the prewriting phase of generating ideas and structuring documents.

The completion phase

In this phase, cosmetic adjustments were made. Issues included selection of font type, font size, indenting of paragraphs and so on. All three groups performed this step similarly, since they were all using conventional software packages. It may be appropriate to mention that cosmetic adjustments were made to some extent during the editing in
the merge/integrate phase. The completion phase is included as a distinct phase to emphasize that towards the end, there existed a period when editing/composing was near zero and the focus was on making the document visually presentable.

Summary of collaborative processes

The overall process of collaboration had many similarities in each case. In each instance the groups went through the four phases of prewriting, drafting, merging/revising and completion. The prewriting phase consisted of generating ideas and planning the overall structure of the document. The ideas generated and the overall structure of the document were evaluated constantly. The structure of the document usually provided discrete segments that could be assigned to different group members for drafting. The drafting phase consisted of composing the document by individuals. The drafts of each segment of the document produced by different group members were merged and integrated. The merged segments constituted the first draft of the document. Each segment of the draft document was evaluated by group members who had not written the segment. The draft document was evaluated for continuity and consistency. Minor editing or adjustments were made. In general, it is possible that at this point new ideas surface or a need is seen for restructuring the document. In the three groups observed, no restructuring of the document occurred. The completion phase involved primarily cosmetic and logistic adjustments to prepare the document for presentation to its audience (e.g. choosing fonts or printing).

The main difference observed between the groups using collaboration technology and the group using conventional technology was in the prewriting phase. Groups, COLL-1 and COLL-2, tried to follow a collaborative model to generate ideas and plan their document, before giving it up to follow a 'scribe' model. Group, CONV-1, used the scribe model from the start. Differences observed between the two groups using ShrEdit included the different way in which the writing task was allocated, and also the difference in the way the merging/integrating task was completed.

The overall use of the collaborative technology was less than anticipated by the researchers. ShrEdit was used in the prewriting phase by both COLL-1 and COLL-2 and in the merge/integrate phase by COLL-1. However, in the prewriting phase, ShrEdit was abandoned in favor of more conventional methods, and in the merge/integrate phase, ShrEdit was used as a single-user tool rather than as a collaborative tool.

In the next section we provide plausible explanations for the limited use of the collaborative software. The available evidence is not adequate to support a single, unequivocal explanation for the limited use, but provides partial support for each of the explanations. The explanations in turn suggest possible issues that need to be considered in the design, and study of collaborative software.

Interpretive analysis of the observations

The primary objective of the interpretation is to seek plausible explanations for the limited use of the collaborative software. We do not address the observed differences between groups COLL-1 and COLL-2, both of which groups were permitted access to collaborative software, but behaved quite differently. We focus on four explanations, and examine the evidence available to support each of the explanations.

Each of the explanations falls within the broader umbrellas of cost benefit or task-technology fit explanations. But within the broader umbrellas, variations are possible.

The global cost-benefit explanation:

The project was a 3-week project. Reward was based on the quality of the project outcome. There was no explicit reward for learning technology or for the extent of use of technology in completing the project. So, the subjects had to intuitively evaluate if costs are lower for circumventing technology or for learning the use of the technology and then completing task. The term learning is used to refer to the process of gaining experience beyond the nominal training in using the software in an effective manner.

Groups COLL-1 and COLL-2 did begin by trying to use the technology in the prewriting phase, but gave up quickly in the confusion that resulted. It is possible that their bad experience with using the technology for prewriting led them to the assumption that the cost of learning to use the technology to perform the task would not be justifiable. Having arrived at that conclusion, they did not use the collaborative technology. So it could be argued that the subjects perceived the cost of learning the use of the software as high compared to potential benefits.

We believe that the global cost-benefit analysis is an important consideration in the decision of subjects to use or not use technology. Faced with a deadline and the uncertain short-term benefits of the software, subjects are likely to save themselves the effort and time involved in learning the software by circumventing the software. So the extent of usage of the collaborative software may be a reflection of an intuitive cost-benefit analysis that is being made by the group rather than the unsuitability of the software for the task at hand.

The cognitive effort explanation

Kahnemann (1973) discusses the tendency of individuals to minimize cognitive effort in performing tasks. When writing is done collaboratively (i.e., prewriting, writing and completion phases are done collaboratively) the cognitive effort required is the effort to generate ideas, write about them, the effort to coordinate the writing process, and the effort to (a) express opinions on other people's ideas (b) incorporate suggestions/ideas of self and others into the overall document. In effect, there are multiple activities that the group members may have to perform simultaneously.
When multiple activities have to be performed simultaneously, it is more likely that the cognitive load is higher than when only one activity is being performed at a time.

The process employed by the groups in this study involved individual writing followed by merging and integrating, i.e., the activities were performed in sequence rather than simultaneously. It would appear that the cognitive effort of the collaborative effort exceeded what group members were comfortable with.

Thus, while collaborative writing may offer some advantages, the cognitive effort required to successfully write collaboratively may bias the users to adopt a sequential process of planning, writing individually and then merging rather than perform each of the activities collaboratively.

The global task-technology fit explanation

When any new technology is developed, it is often developed on the basis of some implicit model of the overall process. For instance, collaborative writing tools make the implicit assumption that the group writing will be done collaboratively, i.e., multiple authors will work on a common document simultaneously. Based on the three groups studied in this project, it would appear that the group writing does not always proceed along that model. An alternate model involves the decomposition of the document into discrete segments, which are written up individually, and then merged. The groups in this study chose the alternate model. Hence, there may be a task-technology mismatch at the global level in the use of collaborative writing tools for group writing tasks.

The local task-technology fit explanation

The process of generating a document by one person involves several steps: generating ideas, planning the document and writing. When the same document has to be produced by a group, there may be additional steps, e.g., merging / integrating. The technology must fit each of these activities. If there is a lack of fit between the technology and the task at any step, problems can ensue. We discuss two types of local task-technology fits: the task-technology fit at a sub-task level and the task-technology fit for the transition between sub-tasks.

First, in the current study, ShrEdit is initially used as a tool for generating ideas and organizing the ideas to plan a document. The groups are overwhelmed by the confusion that results in trying to use ShrEdit for this step, and quickly abandon the tool. This suggests that the features of ShrEdit are not appropriate to support the process of idea generation and organization of ideas by a group, i.e., there is a task-technology mismatch at the local (or sub-task) level. This leads to the further possibility that the group uses this one negative experience to deem the software unsuitable for the remaining activities. So, the groups do not use ShrEdit for other phases such as the collaborative phase of merging / integrating, for which the software may have a better fit.

Second, assuming that the group does not extrapolate the first negative experience with the software to deem it unsuitable for the remaining activities, but is willing to try the software for each of the activities, there is still the problem of transition. The group steps outside the tool to perform the first step (brainstorming in this case) and performs the task either manually or with some other software tool. For the group to return to the collaborative tool in subsequent steps, the product of the idea generation and document planning step needs to be brought into the ShrEdit software from the conventional / manual tools. This transition step is the equivalent of a sub-task. If such transitions are not supported, yet another task-technology misfit exists at a local level.

Summary of analyses: In sum, we argue that there are several possible explanations that could explain the limited use of the collaborative software in the current instance. In any given instance, one or more explanations may be operative. It is not possible to establish the relative significance of each of the explanations in the current case, but it is possible to derive some useful lessons from each, either on the design of groupware or on the process of studying groupware. In the next section, we offer our suggestions.

Suggestions

The analysis allows us to offer some insights on the design of collaborative software and on the process of interpreting results from studies of this nature.

On Design

The insights on design of collaborative software and group support tools fall into three categories: task-technology fit and the issue of peak cognitive effort. The two categories are not necessarily independent.

The task-technology fit: The need for technology to support the specific nature of the task has been argued by several authors (e.g., Easton et al, 1990). The point that we would like to offer is that the fit needs to be considered at two levels, i.e., the global level and the local (i.e., sub-task) level. For instance, in the collaborative writing software, the task is one of pooling effort to produce a common document. The collaborative writing software has to be consistent with the overall task model and be consistent with each of the sub-tasks that have to be performed. In the current case, the possibility that group writing may follow more than one model has been raised. ShrEdit supported one of those models. Thus there was a lack of global task-technology fit in instances when groups adopted a different model for the group writing.

Further, group writing is a sequence of sub-tasks each of which is different in nature. It includes generating ideas, outlining or structuring the documents, writing and so on. It can be argued that the free form collaborative writing tools
time to learn the technology. So single-premature rejection of the technology because it is too much levels, or, the problem of learning curve, which can lead to on the effectiveness of new technology. But single session studies with short training periods on the technology novel effect, which can promote usage above normal studies raise the possibility of either the problem of the spirit, e.g., the use of ShrEdit as a single-user tool in the current study rather than in the intended multi-user spirit. Hence we argue that a collaborative tool must take into account the peak cognitive effort that is being asked of the user and ensure that this effort is within a feasible range.

On Interpreting Studies

Single-session studies (e.g., Olson et al, 1992b; Dourish and Bellotti, 1992) can provide some information on the effectiveness of new technology. But single session studies raise the possibility of either the problem of the novelty effect, which can promote usage above normal levels, or, the problem of learning curve, which can lead to premature rejection of the technology because it is too much effort for the subjects to learn the technology. So single-session studies with short training periods on the technology always leave residual doubt about the generalizability of the results. If the subjects like the technology, was it because it was better or was it because of the novelty effect? If the subjects did not like the technology, was it because the technology was bad or was it because they didn't take the time to learn the technology.

The cognitive effort issue: One of the intents in most group support applications is to permit each of the participants to use their time during a period of interaction most productively. For instance, a collaborative writing tool will allow one individual in a group to continue to generate ideas while another may start the process of structuring the ideas or composing a segment of the document. While this flexibility improves individual productivity, it increases the cognitive effort to required by each individual to keep track of the group's activities. At some point, the expected level of cognitive effort exceeds what the user may be willing or able to put in. At this point, the software support starts to break down. Usually, the support is used in an unintended spirit, e.g., the use of ShrEdit as a single-user tool in the current study rather than in the intended multi-user spirit. Hence we argue that a collaborative tool must take into account the peak cognitive effort that is being asked of the user and ensure that this effort is within a feasible range.

Such residual doubt is usually addressed by conducting longitudinal studies. Longitudinal studies may be conducted in organizations or in the university environment. The university-based longitudinal studies are often constrained by course requirements, i.e., the study must be completed in the span of time available for the project, which cannot exceed a semester. With the many demands on students' time, students constantly balance the effort of learning a new technology to complete a project against completing the project without learning the technology, sometimes choosing the latter. For instance, Zigurs et al (1990) report that 'two groups said they became worried about time constraints and stopped using the system because they didn't have the time to be that organized' (p. 28). So, care is called for in evaluating observations of longitudinal use in the university environment. If the system is used and demonstrates effectiveness in this environment, it can reasonably be said that the system is effective. However, if the system is not used or used incorrectly, then it does not necessarily mean that the system is not effective. Other explanations may be in force. Inferences about the other explanations may be made by analyzing the process of interaction or by debriefing the subjects. However, conclusive claims will be difficult to make. Further studies will be necessary.

Concluding remarks

The design and evaluation of groupware technology is a complex process. No single evaluative test provides conclusive evidence. Researchers must go through a series of iterative loops of design and evaluation to establish the usefulness of the groupware. In the case of ShrEdit, Olson et al (1992a) had observed users performing collaborative tasks without technological support and designed the software on that basis. Subsequently, the software was tested in single-session use and found to be both appropriate and effective (Olson et al, 1992b; Dourish and Bellotti, 1992). Our study, a limited longitudinal study, the results were less encouraging, with little or no use of the software by the subjects. We have offered several plausible explanations. These explanations surface issues that need to be considered in the design of all group applications.

References


Vician, Chelley, DeSanctis, G., and Poole, M.S. (1992). Using group support technologies to support the design of 'Lights Out' computing systems: A case study. IFIP WG 8.2 Working Conference, Minneapolis, MN.

