Group-Based Mediational Leadership in an Online Project Team Context

Edward A. Mabry, Ph.D.

Department of Communication
UW-Milwaukee
Milwaukee, WI 53201-0413
414-229-4371
[FAX] 414-229-3859
eamabry@uwm.edu

Fay Sudweeks

School of Information Technology
Murdoch University
Murdoch WA 6150 Australia
+61-8-9360-2364
[FAX] +61-8-9360-2941
sudweeks@murdoch.edu.au

Paper presented at the annual meeting of the International Communication Association, Communication and Technology Division, San Diego, CA, May, 2003.

Group-Based Mediational Leadership in an Online Project Team Context

Abstract

Mediational leadership focuses on facilitating decision-making and relationship management within a group. This study examined how mediational action could be practiced collaboratively in a multi-group online project team environment to provide expert leadership on a critical task. Data included email achives, questionnaires from group members involved in coding messages for other research, and participant-observation standpoints provided by the authors. Findings showed the expert leaders group was approached for assistance both *directly* and by *looping* inquiries through the group's designated nominal leader. Both types of connectivity to the group appeared effective in providing expert recommendations. The climate of the group also indicated it functioned effectively as a site of mediational action. However, a collaborative model like mediational leadership could prove inefficient for certain CSCW groups or teams due to the amount of time involved in interaction.

Group-Based Mediational Leadership in an Online Project Team Context

Computer-supported collaborative work (CSCW), through the use of virtual teams and groups, has evolved to become a significant facet of organizational life (Scott, 1999). Using teams of workers dispersed over geographical locations and time horizons, yet linked together through access to communication networks, has changed the way people work in groups and redefined the nature of teamwork (Jackson, 1999; Lipnack & Stamps, 1997).

While there has been a substantial amount of research on the overall effectiveness of CSCW (see Scott, 1999, for a thorough summary), relatively little is known about how process management strategies and other forms of leadership activities influence collaborative processes and performance in virtual teams and groups (Pauleen & Yoong, 2001, Yoo and Alavi, 2001). Certain technological forms of CSCW employ nonmember facilitators to implement process management routines and, in effect, function as procedural process leaders (Aukhus, 2001; McGrath & Hollingshead, 1994). Recent perspectives on group collaboration (Stohl & Walker, 2002; Sudweeks & Rafaeli, 1996) have noted that group members' *power* can often be traced to how much control they are able to exercise over shared meanings about collaborative goals or processes as opposed to actual decision outcomes. Virtual teams and groups engaged in CSCW, just like conventional face-to-face interacting groups, clearly need some mechanism for providing guidance and structure in what is typically a relatively unstructured group context.

The study reported here examines an alternative approach to leadership in a collaborative virtual group context. It analyzes the use of a functionally relevant *expert* team as a form of *collaborative leadership* structure. The *asynchronous* communication process of this form of team approach to arriving at guidance on certain task decisions is the focal point of the analysis. The overarching aim of the study addresses how collaborative processes and outcomes

characterizing the project group provides insights into how the construct of collaboration itself can be a useful model for managing collaborative processes.

Group Context

This study focuses on one of the first known instances of a virtual project group being composed to engage in the empirical study of computer-mediated communication in online group contexts (see Sudweeks & Rafaeli, 1996; Rafaeli, Sudweeks, Konstan, & Mabry, 1998; Sudweeks and Allbritton, 1996; for a complete explanation of the project). The project group encompassed over 100 participating researchers from more than a dozen countries on five continents, serving in various capacities at different points in time. The project's goal was to develop a database of randomly selected, content coded, messages sampled from various types of online groups. Project members interacted using asynchronous communication (email) for nearly the entire time the two-year project was conducted. There were occasions when founding members met at conferences and colloquiums or bore the expense of international telephone conversations but these contacts were infrequent and not part of a structured schedule of meetings. Two members involved in the group's inception functioned as coordinating managers.

Over the course of the project's duration, a number of committees and ad hoc groups were formed. Some of these committee groups functioned as standing committees (e.g., Ethics, Copyright). Others were formed in response to specific needs of the quantitative research methodology used in the study (e.g., Sampling, Reliability). About one third of the project's researchers served on one or more of these committees. The committee system was designed by the project coordinators to facilitate decision making. The process of committee formation involved preliminary discussions among all members to clarify issues associated with a particular facet of the research, followed by a call for volunteers. Committees of 3-12 volunteers represented the divergent opinions of the group. After lengthy discussions among committee

members, a policy proposal was drafted by the committee coordinator and submitted to the project group for ratification.

The most unique group formed for the purpose of implementing the research design was a temporary group labeled the *Oracles*. This group was composed of twelve members ("oracles") with expertise in various aspects of research methodology and led by a member playfully dubbed the "Commish" (that is, Commissioner of Oracles; one of the co-authors). The Oracles' responsibilities and procedural process was announced to all project participants. The Oracles group was charged with the responsibility of maintaining the fidelity of data reliability by resolving problems that arose from the application of message coding categories by message coders. When a coder was uncertain of a particular coding protocol, guidance could be obtained by emailing the Oracles via a distribution list specifically created for the group. Email was channeled to the group's coordinator (the Commish) who functioned as a contact person and gatekeeper for the Oracles group. The Commish identified the most likely (although not exclusive) Oracle member to respond to the particular query. When appropriate, the Commish distributed inquiries or perceived problems to the entire group. Additional information on the project and other support groups is available in Rafaeli et al. (1998).

The Oracles group was unique in a variety of ways compared to other groups formed to support the project. First, it had the most narrowly defined task: it functioned exclusively to deal with questions or conflicts that arose from applying message content coding protocols adopted for the project. These protocols were developed collaboratively in the early months of the project. Each researcher participating as a message coder was trained and tested on the use of the coding protocols. Second, the Oracles group's time horizon was interdependent on the coding process, which took place over a six-month period. Third, the Oracles group was responsible for making decisions about work in progress – the coding of messages by researchers (some aspects of the Reliability group's work in protocol training was similar to that experienced in the Oracles

group). Fourth, the Oracles group differed from all other groups in that it was obliged to establish a collaborative relationship with each research coder posing a question for it to consider. Finally, the Oracles group's collaborations were confidential because of the potential contamination effects its discussions could have on the work of other coders and on accurate overall estimates of coder reliability. Coders were discouraged from interacting among themselves while they were engaged in the coding task. Instead, they were instructed to direct questions, decision making quandries about coding protocols, and even their frustrations over the work to the Oracles group.

The key issue addressed in this study is the efficacy of a collaborative form of facilitation, or leadership, within the context of a virtual collaborative research project group. We address two specific questions about leadership facilitation and virtual collaboration: (i) what patterns of group response, if any, characterized how the Oracles group implemented its primary charge? and (ii) what events shaped the group's communicative environment in addition to any patterns that might have emerged in the way it responded to solicitations from coders?

Leadership and Facilitation

The Oracles group represented a form of collaborative teamwork that Stohl and Walker (2002) have conceptualized as a Negotiated Temporary System (NTS). They argue that NTSs are a type of *bona fide* group context. Therefore, they function to mediate collaborative obligations dictated by individual and group goals, constitutive outcomes, and mutual accountability among collaborating partners. Thus, the primary objective of this study is assessing how the Oracles group functioned as an NTS and a source of leadership with respect to the task of message coding.

Barge (1996) has noted that:

> Viewing leadership as a form of mediation shifts from conceptualizing leadership skills solely in reference to task and people problems to couching skills in the language of

> information processing and the strategies associated with acquiring, retrieving, managing, and using information. (p. 322)

Barge also points out that attempting to establish a dialogue between group members is an integral element of the communication skills leaders must embrace in approaching leadership from a mediational perspective.

These group dialogues, though, can take different forms (Barge, 1996), and may vary according to the diversity of group members (Neff, 1995; O'Hara-Devereaux & Johansen, 1994) and fluidity and clarity of group goals and expectations (Shockley-Zalabak, 2002). The Oracles group was particularly characteristic of these contingencies. It was a cross-culturally composed group conceived to support members of the cross-cultural and cross-functional project group from which its members were drawn. The Oracles group's procedures were relatively unstructured and reactive even though its purpose was prescribed. And its primary goal – success in maintaining the fidelity of research data – could not be immediately discerned from its decisions because the work was still in progress.

According to Barge (1996), mediational leadership is not a simple construct. He distinguishes between *object* and *action* mediational leadership. Object mediation involves external information gathering and networking often included under the concept of *environmental scanning.* Action mediation involves more internally focused activities related to achieving particular goals. In particular, action mediational leadership draws on the nature of decision-making functions and relational consequences of decision implementation. The implications of action mediational leadership seem particularly relevant to understanding virtual teams and groups.

Barge (1996) characterizes the attributes of leadership accruing to action mediation practices in the following way:

> The ability of leaders to be effective action mediators turns on two key skills. First, leaders must be able to facilitate group members in decision making, or in selecting from among competing alternatives, that will move the group's goals and tasks. This necessitates leaders being skilled in various communication activities that allow the group to arrive at quality decisions. Second, leaders need to be skilled at managing personal relationships among group members. (p. 324)

Thus, leadership hueing towards action mediation involves facilitating decision-making and relational management. However, it is not clear how a technologically constructed modality of group communication like asynchronous computer communication (Mabry, 2002) supports these leadership dimensions.

Leadership and Virtual Collaboration

Turoff (1991) observed one set of *asynchronous group process factors* that designers of computer-mediated communication (CMC) based group support systems should include – what he labeled *meta individual* and *group processes*:

1. *Regulation:* Managing participation, allocation and ordering of tasks, and process efficacy.
2. *Facilitation:* Organizing, extrapolating, and summarizing information and resources; filtering out irrelevant information; integrating group goals and effort.
3. *Social-Emotional Activities:* Consensus-building; identifying and resolving conflict; promoting cooperation and cohesiveness.

Turoff's (1991) meta process constructs appear analogous to action mediation leadership dimensions. Regulation and facilitation are typically associated with activities that enhance group decision-making, and social-emotional activities involve behavioral requisites of relational development and maintenance.

The construct of decision-making is typically approached in formulaic terms (see, Hirokawa & Salazar, 1999, for a recent summary). However, leadership from a mediational perspective might be better understood with respect to the cumulative impact of leader communication on decision-making (Pavitt, 1999). This makes sense when the more open and formative attributes of group collaboration are viewed as the overarching contextualization of a group's interactive environment (Stohl & Walker, 2002). And research by Poole and Holmes (1995), using structured group decision support systems (GDSSs), has shown that computer-mediated groups are often better able to influence group consensus, perceived decision quality, and decision satisfaction. These results were observed even though computer-supported groups used more than one structurally distinct procedural process path in arriving at a decision.

Researchers studying virtual teams and organizations have attached substantial conceptual and practical significance to *trust* as a factor in the successful performance of virtual groups (Handy, 1995; Javenpaa & Leidner, 1998; Keyworth & Leidner, 2000; McKinney & Mabry, 1999; Shockley-Zalabak, 2002). However, research by Walther (1994; 1997) has shown that relational and group effort perceptions of members in virtual groups relying exclusively on asynchronous CMC are influenced by whether they anticipate having continued interaction with their teammates. Attributions of group trust, interpersonal immediacy, and perceived similarity were positively related to anticipated future interaction in the group.

The preceding discussion has drawn on leadership theory and research framed by the assumption that leadership is a singularly definable role (an *archtype*) that characterizes those who enact it. Fisher (1986), drawing on Weick's (1978) *leader as medium* metaphor, argued that leadership is *systemic*. Leadership is a group's coherence mechanism for organizing systemic action and direction, in dealing with its complexity as a social system. This perspective seems particularly relevant to assessing a collaborative leadership activity like that embodied in using

the Oracles group as a mediational leadership structure. A group dedicated to providing leadership for a complex task and working relationships is arguably in a better position to constructively support and enact the integrative complexity that Weick and Fisher envisioned.

Methods

The aim of this assessment is to draw inferences and conclusions regarding the effectiveness of using collaborative leadership in a virtual project group. The methodology employed in this study might be characterized as a *virtual field* ethnography (Dollar & Mierrigan, 2002). The empirical goal is to explicate how (if) members of the Oracles group constructed collaborative communication both among themselves and other members of the data coding team. According to Stohl and Walker (2002), collaborative processes in NTSs may engage any or all of the following dimensions of group climate-building: decision-making, commitment, trust, power, and knowledge management. How a group manages these characteristics of collaboration should be evident in its communication content and patterns. With respect to leadership through collaboration, the question becomes how these factors emerge to guide research coders and, secondarily, how they affect other committee groups with contemporaneous charges.

Data Forms and Collection

Three distinct forms of *data* inform this project: (1) a log of electronic mail of the Oracles group; (2) information obtained from coders supplied on post-task questionnaires; and (3) the authors' unique perspectives as participant-observers of the Oracles group (one as the "Commish" and the other as a co-founding member of the research project and facilitating participant in some of the group's decisions and discussions). A brief discussion of these data follows.

Email Messages. A log file of N = 210 usable messages (excluding duplicates and misdirected messages more appropriate for other discussion lists provided for the project) was

available for analysis. (Email system and personally identifying information has been removed or abbreviated.) This file was maintained by the "Commish" as part of the Oracles group's working history. Table 1 contains a disaggregation and descriptive analysis of the messages available in the file.

[Insert Table 1 here]

Coder Questionnaires. Not everyone involved with the overall research project served as a message coder. However, approximately one-third, of the over one hundred people connected in some way to the two-year project, volunteered to provide content coding of messages sampled from online groups using the project's standardized codebook. All coders were asked to fill out and electronically return a post-task questionnaire that was available from an FTP server, along with other coding information and the set of messages coded. Access to coded data was restricted initially to members of the research group and, after a period of two years, placed in the public domain.

The questionnaire contained sections on: (a) personal information and backgrounds of each coder, (b) procedures used in coding and transmitting coded data, (c) perceptions of the group and message content coders analyzed, and (d) annotation information about the experience of coding (e.g. noting advice of Oracles affecting coding decisions, technical problems that may have been encountered, etc.).

Coders were both demographically and geographically diverse according to the N = 33 returned questionnaires. The majority of coders were male (57.6%) versus female (42.4%), and, as a group, could be considered *mature* with respect to age with the majority falling into the 36-50 year range (57.6%). Approximately one quarter were in the 21-35 year range (27.3%) and a relatively significant number in the 51-65 year range (15.1%). Also, as a group, coders were internationally diverse although the majority were located in the United States (63.6%). The remaining 36.4% were located in Australia, Brazil, Canada, Israel, Sweden and the UK.

Predictably, coders were well educated with almost half (48.5%) holding a PhD degree and 81.8% reporting holding a Master degree or higher. Most of the members of the Oracles group also volunteered as coders. Therefore, the preceding profile information on coders functions as a proxy for information concerning the composition of the Oracles group.

Participant Observation. As was noted above, the authors were substantially involved in a variety of the project's activities including functioning as the coordinating member of the Oracles committee. However, only the author who co-founded the research project had experience in every phase of the project (e.g., codebook development, coding procedures, message sampling, reliability assessment, etc.). Thus, each author's immersion experience was different and provided them with unique, but complementary, perspectives as analysts of the Oracles group.

Data Analysis

As with any ethnography, data was approached from a descriptive and judgmentally neutral perspective. The corpus of email messages and quantitative data were examined both independently and collaboratively. Due to length constraints, the authors were selective with exemplars used in interpreting findings. These exemplar selections are addressed in the discussion of results. Questionnaire data was submitted to various analyses in order to better understand project and group participant processes.

Results

Surveying the spectrum of data available for analysis led us to the conclusion that findings needed to parallel the sensitizing conceptual framework for the study rather than be bound by its data forms. Thus, the results address the two research questions of interest: (1) collaborative leadership patterns, i.e. the pattern of group response within the Oracles group; and (2) the communication environment of Oracles group, i.e. the events that shaped the group's communication and the way the group responded to solicitations from coders.

Collaborative Leadership Patterns

Two distinct patterns of responsiveness emerged to characterize how the Oracles group reacted to solicitations for assistance from message coders: (1) *direct*, or (2) *looped*. Neither pattern appeared to be a clearly dominant mode of response.

Direct Responsiveness. The pro forma guidance for engaging the Oracles allowed for coders to email queries to the Oracles' distribution list. The Oracles' coordinator identified as the most likely (although not exclusive) channel for group responses. Direct responses involved Oracles who emailed their advice to the coder soliciting it, without looping the message through the Commish (see Figure 1). There was, of course, nothing inherently or normatively improper with this decision.

[Insert Figure 1 here]

In all likelihood, direct responsiveness emerged for two reasons. First, as a means of organizing the group's work, Oracles volunteered to "cover" specific variables defined in the codebook. Thus, each Oracle self-identified with a sense of expertise related to extrapolating certain elements of a message's meaning and was motivated to make that expertise known within the circle of researchers working on the project. Second, as most of the Oracles were also coders, it is reasonable to assume they appropriated a sense of duty in quickly responding to coders needing guidance as they surely would have wanted if the inquiries came from themselves.

An example of direct responsiveness can be seen in the short thread labeled "Using "9" [Missing Data] Code." This thread is also unique in that the person responding was one of the project coordinators who was in the group primarily as a back-up member. His presence, though, was fortuitous in that the question dealt with a generic issue not specifically attached to any message analysis variable, and he was involved with the collaboration that produced the codebook. The message sequence below lists messages in the order they were received

Inquiry to Oracles from Mark:

Dear Oracle(s),

This morning I received a copy of the codebook, 100 messages etc. All seems ok but for one confusing section in the questionnaire:

' 9. Why was it necessary to code 9:

'   MSGNUM:

'   Variable:

'   Reason:

'

'   (repeat for each variable coded 9)

I see no reference to 'coding 9' in the codebook. Can someone please tell me where to look or when coding 9 for a variable is appropriate and what it means. Thanks.

Response from S.R. (project coordinator):

Mark:

You ask:

> I see no reference to 'coding 9' in the codebook. Can someone

> please tell me where to look or when coding 9 for a variable is

> appropriate and what it means. Thanks.

>

The codebook states that you should use 9, very sparingly, when no other answer seems appropriate. About 50 lines down, from first line, it says:

If there is absolutely no way you can code a variable, use 9 (except the 4-column DEPEND2, which will be coded 9999 as indicated). Make a note in the codebook questionnaire (see separate file) for every time you've chosen this drastic measure.

Hope this helps.

Response from E.M. (the Commish):

Mark,

Under the sideheading METHODOLOGY, paragraph 4 [un-numbered], explains that number "9" is to be used as a last resort (sic., "drastic measure") code for any variable that you just can't figure out how to code using the prescribed numeric code options for the given variable. The request of all coders is to make notations of each such encounter (hopefully few and far between) on the questionnaire returned with the coding file.

You are the first coder to point out (correctly I think) that we've buried the explanation for how to escape from a "crash and burn" coding episode pretty deeply in what looks like innocuous general information. Let's hope you won't have to hunt through that section due to any of the messages you encounter coding.

Reaction from Mark:

Thank you Ed. & sorry for asking a daft question. You are the second oracle to point out very kindly that I must be going blind - or daft. I can read but apparently not very well.

The responses to Mark by S.R. and the Commish were posted independently of each other. This was due, in part, to the nature of the inquiry, and the Commish not knowing that S.R. was on the Oracles distribution list. Mark's reaction is also noteworthy. In light of the responses to his question, Mark chose to perceive his question as reproachable. He elected to engage in face-work even though the Commish explicitly noted that the codebook did not contain a clear wording on how the "9" code for missing data could be used.

Looped Responsiveness. The pattern labeled *looping* (see Figure 2) is one that more closely resembles the pro forma procedural process developed by the group and disseminated to

coders. The exemplar reproduced below, labeled “Coding Duplicate Messages”, shows the inquiry being looped through the Commish. The response from the Commish repeated the majority suggestion (accounting for repeated messages using the “Noise” code) along with some contextualizing background that was supplied in messages from other respondents.

[Insert Figure 2 here]

This was an option available to coders but not addressed in the guidelines. Therefore, the inquiry is introduced to the group by the Commish in the form of a message internally reproduced in his message.

Inquiry from Amos (forwarded to the Oracles group by the Commish):

```
I believe this is the first instance of a repost we've been informed
about. Besides his suggestions (appended), I would favor considering:
A.  Recode as NOISE = 1; all other variables as appropriate.
B.  Drop message from sample--with or without replacement.
I don't have a preference between A or B. I would like to arrive [at]
a decision that we can announce to all coders as this is a mechanical
sort of occurrence Oracles shouldn't have to deal with on a case-by-
case basis.
Inquiry from Amos [last name] reads:
Hello:
During coding the 100 messages I had a doubled message: it was
remailed after 15 minutes. Am I supposed to recode it as I first
coded it, to leave it out (as a blank line) or to give '9' to all the
variables?
```

Response from S.R. (project coordinator)

```
> I believe this is the first instance of a repost we've been
informed > about.
> Besides his suggestions (appended), I would favor considering:
```

> A. Recode as NOISE = 1; all other variables as appropriate.

> B. Drop message from sample--with or without replacement.

Ed: I believe your first suggestion is probably best. Granted, each option has a cost attached. My gut feeling is to get the noise information included would be the most helpful.

Response from Oracle N.E.:

I favor just dropping the message from the list, since it simply repeats another (already coded, I assume) message. As to whether or not to replace: Do statistics demand an even 100 messages? If so, I guess would need to replace. If not, I'd favor just leaving that list at 99.

Response from F.S. (project coordinator):

A repeated message is noise and since we are interested in measuring NOISE, I would favour Ed's first suggestion:

> A. Recode as NOISE = 1; all other variables as appropriate.

However, depending on the content of the message and any other information available, either NOISE = 1 or NOISE = 3 would be appropriate. For example, if the message is an announcement of some kind, the author may have intended to send a copy to another list, in which case it should be coded as 1.

Another reason for a repost 15 minutes later would be that the author was expecting an acknowledgement that the message was distributed. The default setting for a list can be no acknowledgement and no repro and if the author is new to the list, he/she would be expecting an indication that the message had been distributed so tried to send it again. In this instance, it could be coded as 3, i.e. intended for the list but not a regular message because it is a repeat. Or there could have been a system problem when mailing and the author wasn't sure the message had been sent, so decided to re-send.

Response from Oracle J.K.:

```
I also recommend coding as noise (with whichever value is
appropriate). It seems to be a very slippery slope if we decide which
messages "belong." Usenet often has multiple postings due to
impatience as well as mistakes, and we should try to capture them.
I vote with "A."
```

This thread contains a number of implications. In some respects it involved overly responsive behavior. Two reactions were from the project coordinators and two others from Oracle group members. As Oracles had already opted for a division of labor in covering inquiries, the multiple responses contradicted the group's nominal structure.

As with the prior example thread (using the missing data code), the substance of this inquiry might also have been the reason for the zealousness of the reactions. The question of how to treat duplicate messages in the sample of messages assigned to a coder did not relate to a particular code variable definition *per se*. It stemmed from potential anomalies related to messaging in the online groups under investigation. Thus, the electronic propinquity of other participants in the research project probably fostered their involvement as the subject of the inquiry posed a unique issue and piqued their interest.

Another example of the looping pattern is found in the discussion thread labeled "Intra-set Message Order." An inquiry was forwarded to the Oracles group by one of the project coordinators because the coder sent it to her instead of sending it to the Oracle group's email address. This is a similar form of indirect access to the group as was evidenced in the preceding example, except the forwarding person was one of the co-founders rather than the Commish. However, both examples might indicate reticence about directly addressing the group for help. The issues raised in the solicitations are equally salient and both warranted expeditious responses.

Inquiry from Ted (forwarded to the Oracles group by F.S. – project coordinator):

This question from Ted is primarily a problem related to the structure of Compuserve messages. I think [S.] is most familiar with this structure - perhaps others may have something to add also?

------- Forwarded Message

Subject: question about data

Fay: I'm finishing the entering of data into a file to be transmitted Monday, but I am concerned about one thing and do not know what oracle to address. Thus to you, oracle of oracles.

My 100 messages are grossly out of order in date - they cover the period from late March to early April, and within that period the dates of messages leap back and forth a lot as I work my way through the messages. In fact, in the list of 100 messages, there are 23 such date inversions, where the next higher MSGNUM has an earlier MSGDATE. This would not worry me, except that often messages early in the set of 100 refer 'back' to messages later in the set (but earlier in date). As a result, I have a major coding dilemma:

1) If I do not catch these inverse orderings, we have data that are incorrect: they _do_ refer to an earlier message but, since I haven't caught the inversion, I do not correctly code the earlier message (later in my set); OR

2) If I _do_ catch the inversion (as I have in some cases), the earlier message coded in the DEPEND variables actually has a higher msgnum.

To put the same point more briefly, the order by MSGNUM is very different from the order by MSGDATE, and this seriously affects the coding of DEPEND1 to DEPEND4. It seems to me to invalidate that coding.

My questions: 1) Is this unusual in these data? 2) _Does_ it create problems of interpretation of the DEPEND variables? 3) Should I be doing anything about it in the coded data? 4) IF my data are unusual in this respect, should I withhold them as contaminating the datafile?

Sorry to be so late in raising this question, but I have only realized the extent of the problem in the last few days.

Response from the Commish:

Ted, Fay, and all:

I had part of the same problem with my list. And, there was a post to Oracles earlier that raised a similar point regarding the date sequence. I didn't have any instances where my message numbers were also confounded with date sequence with respect to coding the DEPENDS variables--at least not that I was able to validate. [I had couple of messages where the connectivity was ambiguous and date chronology was confounded.] I did have to pretty much re-read every message from 1-100 as I was coding so I wouldn't inadvertently do any confounding (which slowed me up quite a bit).

When the question of date sequence was posted to Oracles, [S.] made the point that actual list distributions probably weren't discontinuous even if the dating created that appearance. I noticed with my list that dates flip-flopped but not within what I was perceiving as discrete conversational threads (a tough call at times as I was dealing with no more than a dozen folks on a chat line). Of course, I might either have been luckier than other coders like Ted or just unobservant enough to miss things he didn't.

Ultimately, Ted's point is valid. If he was coding a DEPEND2 as 0060 while coding MSGNUM 0055, I would say his coding was flawed--not due to an error on his part--because one does not expect to encounter FORWARD numbers as the DEPEND2 variable is explained in the codebook.

Under those circumstances, it would appear that his message set would need renumbering and his affixing of message numbers corrected. Actually, I don't know if it's possible to reorder a message sample to resolve this problem. So, there might have to be caveat of sorts included with archive information that flags this sort of problem as "typical" of certain lists.

As for Ted's angst about this dilemma, it boils down to how he resolved the problem for himself during coding. I read his message to say that he inserted numbers for DEPEND2 no matter the order. That's probably the best call under the circumstances because tracking information is preserved. If that isn't what happened, it might be necessary to recode the DEPENDs variables--with or without renumbering messages.

Sorry. This isn't much help. But, neither were these lists or their chronographies. However, I think the problem can be solved and, most likely, without too much additional work for Ted or anyone else.

Response from S.R. (project coordinator):

Fay, Commish, and other Oracles:

Here is what I think. I have not replied directly to Ted -- pending your reactions. Ed: If you agree, can you just forward to Ted?

[Message header removed.]

> My questions: 1) Is this unusual in these data? 2) _Does_ it create
> problems of interpretation of the DEPEND variables? 3) Should I be
> doing anything about it in the coded data? 4) IF my data are
> unusual in this respect, should I withhold them as contaminating
> the datafile?

1) The funny structure of Compuserve messages is common to all Compuserve lists. In fact, the only way to follow the forums in an orderly fashion is to log on VERY frequently. Once you resort to

downloading archives (as we had to), you get the corpora in messed-up order.

2) As long as you return and code later "DEPEND" connections, the variable will not be contaminated. I believe we oracles discussed this problem a while ago, and decided to urge a return to the coding of earlier messages, fixing "DEPEND" codes, if later (by number) messages turned out to be earlier in sequence.

3) Yes, you need to try to catch these dependencies. It makes for more work, but I do not know of a way to tweak Compuserve into spewing the messages in temporal order. Sorry.

4) Definitely DO NOT withhold.

Response from F.S. (project coordinator):

Commish, [S.] and others:

Just a few more words about Compuserve - I think [S.] explained earlier that the messages are archived in thread order, not date order. This is fine except that my limited experience with Compuserve lists is that the archiving is not even in an orderly thread order. I know I found it difficult to find a start date for a sample that made sense.

I agree that the only way for the poor coder to cope with the DEPEND variables is to go back and forth in the sample and record the "last" message referenced no matter where it appears in the sample (previous or forward).

Response from the Commish:

Oracles,

My sense of our discussion of Ted's inquiry is pretty much summed up [in] [S.]'s post. I'll verify that he has received that post (or copy it to him) and add my reassurances.

Fay, please note that Ted may need more time for coding submission as he might have to review his coding of the DEPENDs variables.

I'll take a "no response" as agreement on the above strategy.

Response from Oracle S.K.:

Oracles,

I agree with [S.]'s suggestions. I'm glad I didn't have to code a Compuserve list.

A minor curiousity: Why can't Compuserve be ordered temporally after downloading archives? I'm not a programmer, but wouldn't you just search for a time and then sort chronologically?

Communication Climate of Oracles Group

The foregoing analyses reveals that the Oracles group was quite efficient in its approach to responding to solicitations for guidance. There is also evidence that members were mindful of the tension between responses that would be overly directive (thereby undermining group-wise reliability) compared to responses that might seem unduly vague and unresponsive. The tenor of discussions among Oracle group members was far less circumspect. This serves both as an indication that the group functioned as a self-aware entity, as a genuine social group, and that it became invested in the issues that were brought to it by coders. This deepening and layering of group involvement was evidenced by long deliberations that followed responses to issues raised by coders in their solicitations for guidance. The following excerpts from the "Encrypted messages" thread illustrates the cohesive layering yet critical nature of the communication climate.

From the Commish to Oracle J.K.:

My notes have you covering lines. I pretty much thought that the trailers in question were automatic. I was taken aback to learn that we have imbedded encryptions in messages. However, I didn't glean from your message that transformation of encryption was an expectation of coders. I would have nudged the inquirer to consider only what was uncensored and, then, reflect the question back to

focus on whether or not any codes might cover the situation (and, to my eye, none do).

About right?

Response from S.R. (co-founding coordinator):

If I can throw my hat into this encryption ring, I think we ought to consider the fact that most readers of Usenet have automatic decoders available at their fingertips. The procedure for most people is as simple as typing D or U. Does this change your opinions about how "public" the message is? Also, if we think such messages SHOULD be coded, can we offer the service of decoding them for our coders? (I'll volunteer to do the work, if necessary. But I guess Commish should decide WHETHER to do it).

Response from the Commish:

I'm going to sound like a broken record! If encryption is easy to employ, and known to be commonplace in some groups: (a) why wouldn't there be a standing coding procedure to preempt coder discretion and instruct to decrypt and code? (b) why isn't encryption a code category? Obviously, it's probably too late to add it. However, it's not too late to broadcast a group instruction to decrypt and code all instances of encryption rather than handle it on a case-by-case basis.

An anxious Commish wants to know.

Response from the Commish to explanations from F.S. and J.:

Thanks for bringing me up to speed on the encryption decisions, Fay. Between the explanations from you and Joe, I'm feeling better about the issue. Also, I agree with you that we ought to send "paralell [sic] advice" to coders similarly affected by receiving encrypted items in there sample of messages. There haven't been any additional questions posted about encryption, so I assume it's limited phenomenon. However, suppose there are other encrypted items that

more (less?) experienced coders have "handled" without oracling? Would we want to poll coders and check on just how many instances of encryption are floating around.

Response from F.S.

Ed, rather than polling coders about the frequency of encrypted messages (since most coders probably haven't started yet) why don't we just make a general announcement that encrypted messages should be decrypted and coded (without sounding as recursive as I've just phrased it) and to send a help message to the oracles address if the coder doesn't know how to decrypt.

Response from the Commish:

I guess that is pretty much what I had in mind. It occurred to me that while encryption wasn't assumed to be probative, it was an obvious part of messages that was not addressed either in the codebook or as a coder pre-processing procedural issue. To the extent it's notable in messages on certain lists sampled, I believe we need to have some sort of "official" policy stand on how ALL coders should react to instances where encryption is used.

I would rather keep track of encryption than write it off as it would never be recoverable (in case it re-emerged later as a latent variable of interest).

Three discussion threads (“Encrypted Messages,” “Use of Peripheral Information in Coding,” and “Sampling and Duplicate Messages”), ranging in length from 18 messages (380 lines) to 29 messages (868 lines), involved conflicts over assumptions underlying variable definitions and procedural decisions. The resolutions of these conflicts were normative for the group. In two instances they led to general advisory guidance announcements made to all coders even though message coding was underway (thus, for some coders, prompting a

review of certain coding decisions). The following is excerpted from the guidance announcement for Sampling and Duplicate Messages:

> *From the Commish to project discussion list:*
>
> Over the weekend, inquiries came to the coding Oracles regarding coding a message(s) that occurred more than once in a sample. Oracles' response to this inquiry has project-wide implications and thereby is being posted to all project coders.
>
> A little background. An exact copy(ies) of a message could be included in a message sample for a variety of reasons: (1) mindful, anonymous reposting by group members; (2) accidental multiple transmission by the author; (3) server reiteration--for various reasons; (4) some combination of the above ;-). Additionally, a message might be duplicated in the process of composing the sample. This last option was the subject of some discussion between Oracles and other folks drawing samples.
>
> On the basis of nearly complete information about sampling procedures, we have concluded that it's virtually impossible for sampling to have caused message redundancy in a set of messages distributed for coding. But, if there is any doubt about this point during coding, check message headers for telltale differences in identification terms. And, if you are still in doubt, send copies of messages in doubt to [the Oracles] for assessment.

All of these deliberations became part of the group's mediational history as they were referred to in subsequent assessments of coders' requests for guidance.

As a group, the Oracles were not homogeneous regarding how they individually perceived certain issues. These differences were aired in the background context of the group's interactions, and nothing suggests these deliberations were not constructive. The fact that members did not *leak* contradicting or divisive cues to non-members while they were

functioning in their roles as Oracles suggests that interpersonal relations in the group were generally positive.

Assessing decision-making facilitation must be tempered by the necessary constraints on the directiveness with which Oracles could frame their guidance. The fact that these constraints were known by coders (albeit probably to varying degrees) makes it reasonable to assume being constructively helpful and facilitative was perceived as normative. Moreover, Oracles messages of guidance were most probably characterized by range-bound behaviors that in style—if not in literal verbal expression—were expected by those soliciting guidance. One implicit measure of the group's effectiveness can be seen in the fact that no coder ever reposted an inquiry after a response or asked for clarification regarding a response from the Oracles group.

Discussion

A focal question for this study is whether the requisites of mediational leadership, facilitating decision-making and relational management, could be effectively instantiated in asynchronous virtual groups. An unequivocal response to this question is difficult to provide. In hindsight, it would have been useful to include questions in the post-task questionnaire distributed to coders asking them to assess the Oracles group. However, open-ended items on the questionnaire did not reveal specific criticisms from coders' either about their interactions with the group or any guidance they may have received.

All evidence provided by this study points to the efficacy of using mediational leadership practices in a virtual group context. More importantly, we believe the findings also demonstrate that multi-party direct and collaborative facilitation is both practical and beneficial in CSCW. Further, the study of mediational leadership at a committee level rather than an individual leader level has proven to be an appropriate unit of analysis. The

connectivity afforded in virtual teams and groups appears to be instrumental in making this approach to leadership effective.

We suspect that accessibility and knowledge management applied to the facilitation of task-relevant decisions fosters trust and stabilizes expectations regarding the potential outcomes of collaborative leadership (Jarvenpaa & Leidner, 1998). Member and, by extension, team expertise is generally recognized as a requisite for effective leadership. And, expertise was strongly related to the task of the Oracles group. Trust has also been found to affect the performance of distributed teams (Scott, 1999). Thus, the ways Oracles expressed their guidance to enhance both a sense of confidence in helping coders to reach decisions and conveying empathetic concern for what coders were experiencing suggests a *praxis* and social dynamic consistent with a successful collaborative context (Stohl & Walker, 2002).

This is the outcome Barge (1996) ascribed to leaders as effective action mediators that facilitate decision-making and relational efficacy. Extending action mediation to the *group* level seems reasonable given the findings reported here. Members of the Oracles group facilitated decisions on two levels. First, they assisted coders in arriving at decisions about the coding tasks. Second, their recommendations also facilitated the broader research effort by establishing validity checkpoints associated with the research methodology.

Findings regarding relational management issues were less explicit in the data but were suggestive. In both their dealings within the Oracles group and with researchers soliciting their advice, members of the Oracles group appeared to relate constructively and with a minimum of disagreement or conflict. There were no instances of destructive *affective* conflict involving the Oracles group.

At first blush, the notion of leadership emanating from a unitary group or team for the benefit of other individuals or groups seems far-fetched. Group and organization scholars

typically think of leadership as a property of a group and not as a group's beneficial task or service. There is some precedent for investing a group with the task of leadership. It is not uncommon for organizations to compose an "Office of the President" in which two or more senior executives share leadership responsibilities. This is often done as part of succession planning and as a precursor to selecting a new chief executive officer.

Investing a group or team with the role of leadership is not a common practice. And, even the Oracles group is a rather unique instance of group-based collaborative leadership. It evolved as a specific consequence of the needs of the research project group given its tasks, size, and history. A single expert person as a resource for the coding task, given the potential demands on the person and consequences of their decisions, did not seem well-suited to the perceived needs of the overall project group. A collaborative effort seemed more contextually appropriate.

The extent to which this model can be applied to other CSCW groups or teams is less clear. From a purely operational point of view, particularly when looped responsiveness is expected (or emerges as a modal pattern of response), a group collaboration model of leadership might not prove to be very efficient. Mediated group communication is typically less time-effective than F2F interaction (Scott, 1999). Therefore, the more online interactants that are involved in mediational action the more time-consuming the collaborative process will be. The question that would always need to be answered before initiating this type of leadership process is whether the quality of outcomes it can provide can offset the greater communication demands and reduction in efficiency that might be unavoidable in employing a collaborative leadership model. The experience using the Oracles group in a complex, multiple role and multi-group context clearly answered this question in the affirmative.

References

Aakhus, M. (2001). Technocratic and design stances toward communication expertise: How GDSS facilitators understand their work. *Journal of Applied Communication Research, 29*, 341-371.

Barge, J. K. (1996). Leadership skills and the dialectics of leadership in group decision making. In, R. Y. Hirokawa & M. S. Poole (Eds.), *Communication and group decision making, 2nd. Ed.* (pp. 301- 342). Thousand Oaks, CA: Sage.

Dollar, N. J. & Merrigan, G. M. (2002). Ethnographic practices in group communication research. In L. R. Frey (Ed.), *New directions in group communication* (pp. 59-78). Thousand Oaks, CA: Sage.

Fisher, B. A. (1986). Leadership: When does the difference make a difference? In, R. Y. Hirokawa & M. S. Poole (Eds.), *Communication and group decision making*, (pp. 197-215). Thousand Oaks, CA: Sage.

Handy, C. (1995). Trust and the virtual organization. *Harvard Business Review*, (May-June), 40-50.

Hirokawa, R. Y. & Salazar, A. J. (1999). Task group communication and decision-making performance. In L. R. Frey (Ed.), D. S. Gouran, & M. S. Poole (Assoc. Eds.), *The handbook of group communication theory & research* (pp. 167-191). Thousand Oaks, CA: Sage.

Jackson, P. (1999). *Virtual working: Social and organizational dynamics*. London: Routledge.

Jarvenpaa, S. L. & Leidner, D. E. (1998, June). Communication and trust in global virtual teams.

Journal of Computer-Mediated Communicaton, 3. Retrieved August 31, 2001, from http://www.ascusc.org/jcmc/vol3/issue4/jarvanpee.html

Kayworth, T. & Leidner, D. (2000). The global virtual manager: A prescription for success. *European Management Journal, 18,* 183-194.

Lipnack, J. & Stamps, J. (1997). *Virtual teams: Reaching across space, time, and organziational boundries*. New York: John Wiley.

Mabry, E. A. (2002). Group communication and technology: Rethinking the role of communication modality in group work and performance. In L. R. Frey (Ed.), *New directions in group communication* (pp. 285-298). Thousand Oaks, CA: Sage.

McGrath, J. E. & Hollingshead, A. B. (1994). *Groups interacting with technology*. Thousand Oaks, CA: Sage.

McKinney, V. & Mabry, E. (1999). Online collaboration of groups with varying tasks and goals. *Proceedings of the Fifth American Conference on Information Systems, USA, 5,* 352-354.

Neff, P. J. (1995). Cross-cultural research teams in a global enterprise. *Research-Technology Management, 38,* 15-19.

O'Hara-Devereaux, M. & Johansen, R. (1994). *Global work: Bridging distance, culture, and time.* San Francisco: Jossey-Bass.

Pauleen, D. J. & Yoong, P. (2001). Facilitating virtual team relationships via Internet and conventional communication channels. *Internet Research: Electronic Networking Applications and Policies, 11*, 190-202.

Pavitt, C. (1999). Theorizing about the group communication-leadership relationship: Input-process-output and functional models. In L. R. Frey (Ed.), D. S. Gouran, & M. S. Poole (Assoc. Eds.), *The handbook of group communication theory & research* (pp. 432-472). Thousand Oaks, CA: Sage.

Poole, M. S. & Holmes, M. E. (1995). Decision development in computer-assisted group decision making. *Human Communication Research, 22,* 90-127.

Rafaeli, S., Sudweeks, F., Konstan, J. & Mabry, E. (1998). ProjectH oveview: A collaborative quantitative study of computer-mediated communication. In F. Sudweeks, M. McLaughlin, & S. Rafaeli, (Eds.), *Network & Netplay: Virtual groups on the Internet* (pp. 265-281). Menlo Park, CA: AAAI Press/The MIT Press.

Scott, C. R. (1999). Communication technology and group communication. In L. R. Frey (Ed.), D. S. Gouran, & M. S. Poole (Assoc. Eds.), *The handbook of group communication theory & research* (pp. 313-334). Thousand Oaks, CA: Sage.

Shockley-Zalabak, P. (2002). Protean places: Teams across time and space. *Journal of Applied Communication Research, 30,* 231-250.

Stohl, C. & Walker, K. (2002). A bona fide perspective for the future of groups: Understanding collaborating groups. In L. R. Frey (Ed.), *New directions in group communication* (pp. 237-252). Thousand Oaks, CA: Sage.

Sudweeks, F., & Rafaeli, S. (1996). How do you get a hundred strangers to agree: Computer mediated communication and collaboration. In T. M. Harrison & T. D. Stephen (Eds.), *Computer networking and scholarship in the 21st. Century university* (pp. 115-136). Albany, NY: SUNY Press.

Sudweeks, F. & Allbritton, M. (1996). Working together apart: Communication and collaboration in a networked group. In C. D. Keen, C. Urquhart & J. Lamp (Eds), *Proceedings of the 7th Australasian Conference of Information Systems (ACIS96),* Vol. 2 (pp. 701-712). University of Tasmania, Australia: Department of Computer Science.

Turoff, M. (1991). Computer-mediated communication requirements for group support. *Journal of Organizational Computing, 1,* 85-113.

Walther, J. B. (1994). Anticipated ongoing interaction versus channel effects on relational

communication in computer-mediated interaction. *Human Communication Research, 20,* 473-501.

Walther, J. B. (1997). Group and interpersonal effects in international computer-mediated collaboration. *Human Communication Research, 23,* 342-369.

Weick, K. E. (1978). The spines of leaders. In M. McCall & M. Lombardo (Eds.), *Leadership: Where else can we go?* (pp. 37-61). Durham, NC: Duke University Press.

Yoo, Y. and Alavi, M. (2001). *Computer-mediated communication and emergent leadership in distributed teams* (Working Paper). Case Western Reserve University, Cleveland OH.

Table 1.

Summary of Oracles Group Discussion Threads

Theme/Issue Label	# of Msgs	# of Contri-butors	# of Msg Lines*	Time (days)
Organization of Oracles Group				
Orientation of Commissioner	12	2	612	2
Commissioner solicitation of Oracles	29	16	488	16
Confirmation of Oracle members	6	6	282	6
Oracle procedure setting	24	8	881	8
Oracles Group Responses and Deliberations				
Message sample sequencing	2	2	24	1
Oracles discussion: encrypted messages	29	8	868	4
Interpreting subject lines	5	4	197	3
Message content codes unclear	6	4	325	4
**Using “9” [missing data] code	4	3	34	1
Information used discerning gender	2	2	28	2
Oracle discussion: use of peripheral information in coding	26	7	788	7
Noise/nature code interpretations	9	4	315	4
**Coding duplicate messages	5	5	48	2
Oracles discussion: sampling and duplicate messages	18	8	380	4
Coding all quotation messages	3	3	40	3
Clarification of category meanings	7	5	260	5
Coding message FORMAT variable	2	2	48	1
Submitting coded data	6	5	27	2
**Intra-set message order	6	5	137	2
Inquire on AUTHORID variable	2	2	32	3
New coder questions on codebook	7	5	270	3

*Message lines measure excludes header and footer lines.

**Denotes messages included in text for analyses.

Figure Caption

Figure 1. Direct message response from an Oracle.

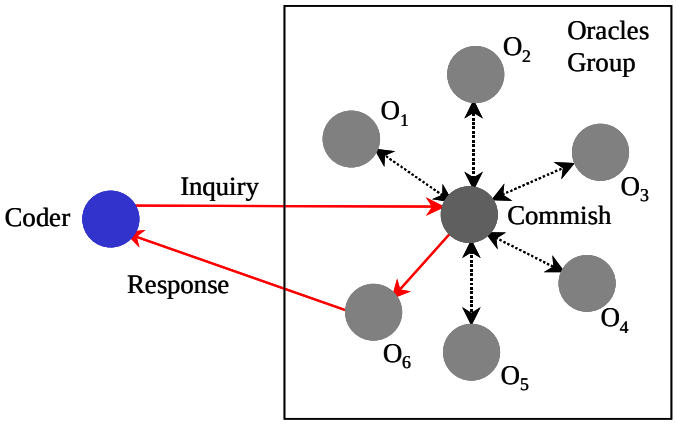
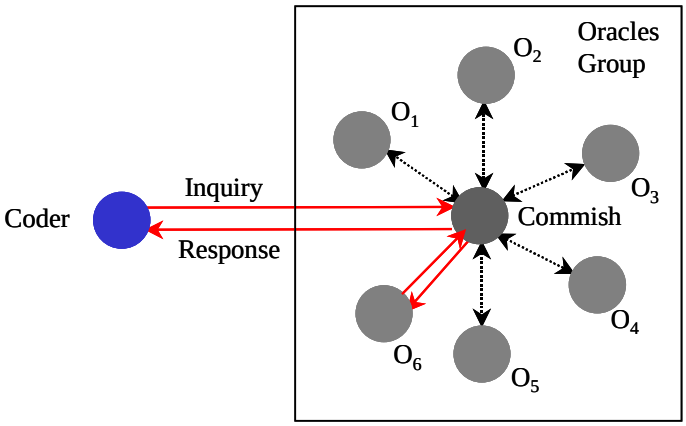

Oracles
Group
O2
O1
O3
Inquiry
Coder
Commish
Response
O4
O6
O5

Figure Caption

Figure 2. Message looping through the Commish


Oracles
Group
O2
O1
O3
Coder
Inquiry
Commish
Response
O4
O6
O5