JOURNAL OF ORGANIZATIONAL COMPUTING, 1, 85–113 (1991)

Computer-Mediated Communication Requirements for Group Support

Murray Turoff
New Jersey Institute of Technology

This article presents an overview of the historical evolution of computer-mediated communication (CMC) systems within the context of designing for group support. A number of examples of design features to support specific group tasks are illustrated. The result of this is the synthesis of a number of observations on the assumptions and goals for the design of CMC systems. An emphasis is placed on the advantages offered groups by asynchronous support of the communication process, self-tailoring of communication structures by users and groups, and the integration into the communication system of other computer resources and information systems. The systems that have been developed recently at New Jersey Institute of Technology (EIES2, TEIES, and Personal TEIES) are used to illustrate the translation of design objectives into specific features and functions.

computer-mediated communications, groupware, GDSS, CSCW, computerized conferencing, message systems, electronic meeting systems, hypertext

"The mighty telescope looks afar.

But finds no place to park a car."

—Samuel Hoffenstein, *Pencil in the Air*

This article was originally an invited paper for the Conference on Organizational Computing with respect to Theories and Technologies for Computer Supported Work, held at the University of Texas, Austin, November 1989.

The work on the EIES2, TEIES, and Personal TEIES systems described in this report has been made possible by funding from a variety of sources, including the Annenberg/CPB Project, IBM, AT&T, Hewlett Packard, the N.J. Commission on Science and Technology, the N.J. Office of Telecommunications and Information Services, and the New Jersey Department of Higher Education.

Also, acknowledgment is due to Starr Roxanne Hiltz, James Whitescarver, and John Foster for their contributions to the thoughts in this article.

Correspondence and requests for reprints should be sent to Murray Turoff, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ 07102.

1.0. HISTORICAL PERSPECTIVE

This section reviews some of the history of computer-mediated communications (CMC) to support group-oriented activities. It highlights some of the design concepts that are fundamental to the evolution of the use of computers to facilitate group communications.

1.1. Delphi Conference

The first effort at creating a specific computer system designed to asynchronously enhance group problem solving was a computerized version of a Policy Delphi process [39], created in 1971. Delphi is a methodology often used for complex, unstructured problems, such as trying to predict potential breakthroughs in biomedical research and their societal impacts. The objective of a Policy Delphi is to develop the strongest pro and con arguments for alternative resolutions to a policy issue (e.g., a Hegelian Inquirer [4]). This first computerized conferencing system allowed the members of the group to track suggested resolutions of a policy issue; to vote on the alternatives, using scales for "desirability" and "feasibility"; to make pro and con arguments associated with those resolutions; and to vote on the arguments, using dimensions of "importance" and "validity."

The recent gIBIS system [5] was a replication of this type of communication structure, with the addition of graphics. Although gIBIS was classified as a Hypertext system, all Delphi designs incorporated explicit linkage structures tailored to the types of objects of discourse (nodes). In fact, all the conference systems that have ever been built can be viewed as incorporating a tailored Hypertext structure [15].

Often misunderstood, Delphi's basic objective is the design of communication structures (usually with pencil and paper) to coordinate information exchange in asynchronous problem-solving groups. Early attempts at computerization of alternative Delphi designs led to the following observation [27, pp. 490–491], which remains valid:

> In practice, one should view computerized conferencing as the ability to build an appropriate structure for a human communication process concerning a specific subject (problem). . . . If the individuals enter such a discussion with fake names, then we have defacto a "Delphi Conference." The computer allows us to go from the complete Delphi mode to various mixed modes such as that in which the conferee is able to decide whether an individual comment is signed with the person's real name. It is the view of the editors (Linstone and Turoff) that the question of anonymity or its degree is less crucial to the definition of Delphi than the concept of designing the human communication structure to be used. The hundreds of meaningful paper-and-pencil Delphis that have been done represent a storehouse of knowledge on the design of human communication structures for implementation on modern computer communications systems.

1.2. EMISARI (Emergency Management Information System and Reference Index)

EMISARI was developed at the Office of Emergency Preparedness in 1971 [15, 24, 28, 31, 51] to support decision making during declared national emergencies. Its functions included messaging, conferencing, and data reporting in crisis management situations. It was first used in the wage-price-freeze as a crisis management information system. For over a decade, it was used on some 20 or more declared federal crisis situations, such as commodity shortages (e.g., chlorine, natural gas, and oil) and major strikes (e.g., transportation and coal).

EMISARI provided the ability for 100–200 people scattered around the country to track, interpret, and reorganize both the qualitative and quantitative information associated with a rapidly changing and unpredictable situation. It had many significant features for collaborative group processes that have not yet appeared in any commercial systems. Some of the features that relate to group support are listed next.

- The system monitor could establish a variety of data forms and then assign to particular system members the responsibility to fill in those forms. The resulting data base identified the human sources of the information, and when they had last updated the data.
- People could address messages to the "data," so anyone trying to interpret the data, explain why it was peculiar, or act on it, could associate a discussion (set of footnotes) with the data. People retrieving the data would be informed of the associated discussion. As a result, group reporting activities were self-organizing.
- One type of data table was a time series that calculated a regression extrapolation each time a new entry was made, and flagged values outside of expected predictions.
- Group notebooks (bulletin boards) had separate roles for those maintaining entries and those able to only read. Tracking of key words searched in these entities determined what people were looking for but could not find. This was used as an indirect communication channel to the group that made decisions about new types of information that should be added to the system.
- A directory of members that automatically accumulated the responsibilities of members, with respect to reporting data and the control of various conferences and notebooks.
- A tracking data format enabled people to establish the sequence of steps a "case" or "problem" would go through, and who was responsible for the determination of the outcome of each "step" in the process. As steps were completed, the system alerted those involved in the process to their turn to deal with the case. Anyone could retrieve a case, the current status and actions taken to date, and who had responsibility at the moment.
- A simplified integrated interface to a multiregional input/output data base allowed people to get projections of the impacts of changes in industrial

> output due to disruptions in the flow of goods, services, and commodities. This was a clear example of the integration of communications, models, and data bases as part of the CMC system.

The most important aspect of this system was that the structuring of data reports, roles, and responsibilities could be modified dynamically within a few hours to adapt to changing requirements. Furthermore, it allowed the individuals to self-organize the groups involved and the routing of communications, based on the content of what was being communicated. In a crisis situation, the desirable composition of a group to address a particular issue or problem is entirely unpredictable. This was a clear example of the ability of CMC systems to allow the "content to be the address."

Approximately one-third of the software in EMISARI was devoted to supporting tools for specialized human roles that were an integral part of the design. This concept of building human roles into the design has been common, to various degrees, to most other computerized conferencing systems built since that time.

EMISARI was a system that allowed well over 100 people working 12–16-hour days over periods of months to coordinate their activities and collaborate on the analysis of pending decisions. It served, in every sense, the same functions that have been ascribed to real-time situation rooms.

In the business world, the nearest equivalent to the original EMISARI is a version of EQUAL, developed by David Morris. It is used internally by IBM to handle external information requests and problems from the field about a wide range of products. It highly structures the roles people may play in reacting to such requests, the types of reactions that can be made, and the status of these reactions. EQUAL is used by about 30,000 people in IBM. One of the most interesting innovations in EQUAL was the first internal design of a set of fundamental privileges that could be used to tailor any resulting role. This is a key concept in our new systems at the New Jersey Institute of Technology (NJIT).

Another example of tracking oriented to this class of systems is MONSTR (Monitor for Software Trouble Reporting), which was highly structured around the group activity of tracking and responding to problems, and allocating resources to deal with maintenance of software products [3].

The common thread for all these systems is that the software dealt only with human roles, communication protocols and structures, data structures, and text item types and status. There was nothing built in that dealt with content or specifics of a particular problem. This is counter to much of the early paradigm surrounding various "groupware" efforts. The design of a communication system should be content independent. The design should focus on process.

1.3. MAILBOX

MAILBOX was developed by Scientific Timesharing and I. P. Sharp in the late 60s. On the surface, it was designed as a detailed replication of the post office. It contained such features as certified and registered mail with return confirmations. It was used successfully by management to run a decentralized operation of the company. One of the key features leading to its success is that the mail was kept in a centralized file for a sufficient period so that all updates, changes, and related mail items by members of the management group could be tracked and organized around the pertinent topics and projects. Although MAILBOX utilized the mail metaphor, the need for a "group memory" to service organizational requirements was an integral part of the design.

1.4. FORUM

FORUM was developed in the early '70s by the Institute for the Future (IFTF) [48]. It was, at first, a rich system for doing Delphi exercises. Then a very simplified version was developed, which was very similar to the linear transcript concept in the EMISARI subsystem "conference." The objective—a good one—was to study the impact of this form of communication in its simplest form for making relative comparisons with other communication mediums. As a result, IFTF did accomplish significant work in the area of evaluating this technology relative to other alternatives. In later years, it was licensed to Info-Media for commercial use. The current NOTEPAD system is a derivative of the earlier FORUM system.

HUB was a later effort of IFTF to provide a system for group model building, which had many interesting concepts and did receive some test and evaluation treatment. Unfortunately, it was ahead of the availability of personal computers, which make the concepts far more viable for real utilization. Graphics, and the local editing and browsing in composite documents, is really the key to success in this area.

1.5 CONFER

CONFER was developed at the University of Michigan under the guidance (as thesis adviser) of Professor Merril Flood [53]. Flood was initially interested in group preference and decision functions. The initial version of CONFER allowed for some very sophisticated group decision processing. One of the group response or vote structures essentially was to append replies directly to a new comment at the time it was being viewed. This became the main theme to the use of CONFER, and in the final system that went into wide-scale use, most of the ideas related to preference voting were not used. CONFER became the first system in which the transcript was organized by replies rather than linear in time. This has proven to be a very popular alternative that is reflected in later systems. However, it did not take long to observe that, given a system with time-sequential transcript organization and one with reply organization, comments would be a third the length in the reply-oriented systems. This is because the normal interaction mode in the time-sequential system is the first to view all waiting new items in a conference before responding. In the reply orientation mode, user interaction involves looking only at replies to a given item. The sequential organization seems to stimulate more reflective, longer comments, and the response structure seems to stimulate top-of-the-head responses on specifics. Both have their benefit for specific communication objectives. CMC systems must allow for both options, and the ability of users to adapt the system to their cognitive style of communicating.

1.6 EIES (Electronic Information Exchange System)

EIES was designed in 1975 and became operational in 1976. It was the first system designed for the specific purpose of exploring this application of computers and advancing the state of the art by prototyping new developments. EIES, to the new user, presented a number of capabilities: messages, conferences, notebooks, and a directory of members. Notebooks were private document composition spaces, but allowed for joint authorship and editing of one another's material. Also included was a language (INTERACT) that functioned as an interpreter on top of the hardcoded system. The INTERACT programs could be stored in any text item and executed with the privileges associated with the given text item. INTERACT could act on any data passed between individual users and the basic system. As a result, it was a very powerful tool for prototyping new facilities on a systemwide, selected group or individual user basis [47]. For the controlled experiments on human problem solving conducted on EIES, with INTERACT it was possible to provide unique interfaces for the subjects of the experiment [13]. Another key feature of EIES was a fairly extensive activity monitor that provided detailed data on system usage for evaluation purposes. The following sections present some examples of specific tailoring for group support that occurred on EIES.

1.6.1. Terms. The Joint Electron Device Engineering Council (JEDEC), under the aegis of the Electronic Industries Association, used EIES from 1978 to 1980 (when National Science Foundation funding for the project expired) for selected aspects of its work of promoting hardware and software standardization in microcomputer/large-scale integration products. JEDEC's standardization activities are conducted by a series of numbered committees, which ordinarily communicate only through quarterly face-to-face meetings, with phone and/or mail in between. A great deal of JEDEC's work in reaching standards that will obtain the required unanimous approval of its members involves first reaching agreement on a set of terms and definitions that apply to a given standardization topic. In addition to free-form discussion in conferences, a structured decision aid was programmed to support the process of developing and reaching agreement on such a standard set of terms and definitions. Called "TERMS," this system allowed any member of a committee to:

- Add a proposed term (abbreviation and full name) to a list.
- Add a proposed alternative definition anonymously to the list.
- Make a comment about the desirability of a given definition.
- Vote on each proposed term and definition.
- Revote at any time, based on current votes and new alternatives.

Straw votes could be entered anonymously, and tallies of all votes on all proposed definitions were always available to members. The set of items contained in the TERMS subsystem for any specific standards effort was called a glossary and was maintained as a separately organized data base, associated with the group's conference. The TERMS system was designed and redesigned with the participation of interested project members [22].

No doubt, one of the important aspects of this system was the ability to make anonymous or pen-name comments and to vote without anyone knowing an individual's vote. This helped considerably in overcoming the concern of some members about giving away leads or indicators to current R&D developments in the companies they were representing. Although JEDEC committees were "collaborative" groups, they were not necessarily "cooperative" ones with respect to all the group objectives.

There has been a large number of similar applications in the medical field [26]. The effort by a group of 12 world-renowned researchers in hepatitis to arrive at an updated knowledge base for practitioners required unanimous agreement on wordings taken from the research literature [2, 34]. This group was examining all the recent research literature and regularly collaborating to reach agreement via computerized conferencing.

It is interesting to note that the TERMS software was also used in an informal group conference (TRANSFORM) that was devoted to discussing "life" views and experiences. The sorts of terms on which members of the group contributed viewpoints were "vision" and "love." This serves to illustrate that a well-designed group communication structure can serve a wide variety of applications. In such conferences, one has seen the advantage of anonymity in allowing people to discuss suicide or near-suicide attempts on their part. Furthermore, in most Delphi experiments with groups, the change rate when one is not identified with a comment or vote is usually more than 30%. This is much higher than in comparable face-to-face meetings. The resiliency of this form of communication is illustrated by its application to group therapy processes [32, 33].

1.6.2. TOURS and Hypertext. Hypertext functionality is a natural requirement in any CMCS. In the process of communication, there is a need to refer in a message or comment to other items of text that occurred in earlier communications. EMISARI, for example, used a General Markup Language (GML) approach to provide "virtual referencing" of existing text items. A user could virtually place items written by others in his own text, and could also create links to items to be updated later by others. This meant that a report could be updated automatically as people made changes to their contributions. Also in EMISARI, the moderator had the power to link people to the reports to which they were to contribute. One might have characterized this as a "hypergroup" system. The fact that people and material are linked in many different roles is another way of characterizing CMC systems.

This "hyper" quality of CMC systems is best illustrated with a subsystem designed on EIES called TOURS. TOURS was developed by Peter and Trudy

Johnson-Lenz [21] to encourage comprehensive planning among managers. It allowed a complete network of "text items" but had some very unusual and interesting features not found in most other early examples of Hypertext.

A node in this system had a number of parts:

- A text scenario about a particular aspect of the overall planning problem.
- A set of votable issues that everyone reading the material could vote on.
- A set of comments made by everyone who stopped at this node in their travels through the network.

Other key features of this system were:

- "Travelers" through the network of linked fragments could record or modify their assessment of the importance of a set of fundamental factors related to the overall problem.
- An artificial persona, a guide named Joan, presented possible travel stops that the travelers could make based on where they had been, the factors the travelers were rating, and the conditions set up by the creators, or "weavers," of the original network.

The underlying tour network could be structured by the "weaver" of the tour. At one extreme, the structure started travelers at a very specific focused position and the network fanned out into many diverse related topics. Or, the tour could be structured to guide users to a very focused decision-oriented issue after having started from very diverse, loosely coupled aspects of the overall problem.

TOURS was one of the few early implementations we have observed of a Hypertext functionality that seems to embody what Mitroff et al. [29] characterized as a "Myth Information System," in that it allowed the weavers and users to contribute to the lore and wisdom of the current situation and to adapt the nature of the network and its content based on those contributions. Both the voting process and retrieval behavior influenced the guidance process for exploring the resulting web. The combination of Hypertext and CMC leads to an ability to provide collaborative development of the linkages, and this is what is necessary to deal with the problems of equivocality and uncertainty of information in an organizational setting [6].

"Question" is a feature developed initially for our work in designing a Virtual Classroom™ [12] in the CMC environment. It utilizes a comment to pose a question, and answers that are attached to the question by a conditional link. Each conference member is prevented from seeing the answers that have been generated as subcomments until he or she first supplies an answer. Question serves to illustrate a conditional structure for the delivery of new material to participants in a group. Most faculty members who have used the Virtual Classroom™ software tools, have felt that this feature is key to the findings that it is more effective in some way than being face to face in the classroom [12]. A similar structure may be used in any group process to encourage each member to bring up his or her ideas independent of the other members of the group.

1.6.3. Other Examples. Over 25 special subsystems were developed on EIES over the period 1976–1987. In addition to those mentioned earlier, the following are worthy of note:

- RESOURCES is a data-base subsystem that allows for the mixing of qualitative and quantitative information in specified formats for both creation and retrieval. The unique feature of RESOURCES is that it was designed to handle the problem of a group of individuals gathering and validating entries to a data base. Therefore, all entries are identified by who is contributing them, and it is possible to associate comments with any entry in the data base.
- REPORTS is a group-oriented authoring system that provides the ability to establish outlines of a report and signify different roles for individuals at every major point in the outline. Therefore, one can assign authorship, editing, contributing, and organizing privileges to different individuals at different places in the planned report. This system followed many ideas in Englebart's work [8, 9] in collaborative composition and added very explicit role structures for editing, organizing, and producing the resulting documents.
- TOPICS [45] was designed to handle unpredictable information exchange where a large group of people (100 or more) were involved in trying to exchange information. Every member of the network was allowed to send three line inquiries to every other member. Each recipient of an inquiry could select whether to track future responses to the question. Responses could be supplied by any member, but also had to be limited to one page. There was a human indexer who had the software-supported power to keep all the keys consistent. A human editor had the job of collecting the responses, eliminating duplications, and developing summary briefs. The inquiries, responses, and briefs went into a data-base structure for later retrieval. A simplified version of TOPICS became PARTICIPATE on the SOURCE.
- MARKETPLACE [41] was a system to allow people, within the context of a CMC system, to buy and sell information among one another. It had some unique features, such as allowing the purchaser of an information item the ability to attach their consumer reaction to the original advertisement, and this consumer comment could not be removed by the seller.
- SURVEY is a complete system for doing surveys, and allows those taking the survey to get summaries of the results. In one specialized evaluation application, we put up a standardized psychological test and provided the user an assessment of their personality type. For our purposes, these data were correlated with the activity data on their usage, to look at correlations with psychological profiles and system usage.

- GDSS for Zero-Based Budgeting: One of the most important systems built as part of EIES was an asynchronous version of Zero-Based Budgeting (ZBB) [1]. ZBB has largely been a failure in the face-to-face mode, because of the huge amount of communication requirements it normally adds to the already top-heavy budgeting process. The CMC was utilized quite successfully by about 30 people to allocate $1.5 million of capital budget funding without a single face-to-face meeting of the group. The complete design and field trial evaluation is in the thesis report. It demonstrated that an approach to budget planning that had difficulties being used in the face-to-face mode could be better utilized in the CMC asynchronous mode.

The key observation about all these specialized structures on EIES is that they went through considerable design evolution. Both user participation in design conference discussions and feedback from the users of the prototypes were used to tailor the structures to the needs of the users [14, 23, 44, 47]. Another important finding was that it was possible to use the technology as a tool to better understand individual and group problem-solving processes [16].

2.0. ASSUMPTIONS FOR AND GOALS OF CMC SYSTEMS

Based on almost two decades of activity in the design and evaluation of CMC systems, the following conclusions are offered on desirable goals and objectives for future research and development for CMC, GDSS, Groupware, CSCW, EMS, Coordination Systems, or whatever jargon one wishes to use:

- The objective of a CMC system is to provide an opportunity for a group to exhibit "collective intelligence" [15]. This means that the results of the group communication process are better than the result any single member of the group could have obtained alone. Collective intelligence is a very significant objective, because it is measurable in real situations and across the use of different communication alternatives.
- A CMC system needs to support group communications 24 hours a day and offer the flexibility of being used synchronously or asynchronously [42]. Individuals do not deal with problems only when they meet together as a group. Nor do they operate, in most situations, as only one group. In real organizations, groups are very fluid in their nature, and the process is one of overlapping and intersecting subgroups.
- The benefits offered by CMC result from the ability to utilize the computer to tailor the communication structure to fit the nature of both the application and the group [38].
- Appropriate communication structures are extremely sensitive to group norms and organizational culture [11].
- Individuals have a great deal of leeway in organizations as to what mode of communication they will use for what purpose. Individuals and groups cannot be ordered to use CMC technology [11].

- The same interface that allows individuals to communicate with other individuals can also allow individuals to communicate with other computer and information resources. A communications-oriented interface is an ideal systemwide interface for accessing data bases and running analysis routines [38, 46].
- Both individual and group problem-solving requirements imply that one must integrate computer resources as part of the communication process for many real-world problem situations [38].
- Individuals and groups must be able to exercise a high degree of selective tailoring in using CMC systems [45].
- Groups evolve their use of the CMC systems, and the development of new facilities must be part of a planned feedback process [44].
- Human roles, and the computer support of human roles, are key factors in the success of group activities [15]. Providing system privileges that reflect the flexibility of human communication processes is essential to supporting human roles.
- The privacy and security (as well as reliability) of human communications are essential to the acceptance of the system [40].
- Information overload and the ability to deal with large user populations and large group size are the driving forces for the appropriate design of CMC systems [14].
- There are numerous difficult design trade-offs, in terms of conflicts in control and structure of communications. These conflicts revolve around differences in individual and group behavior and objectives in the communication process [40, 43].
- The appropriate metaphor in CMC systems for experienced users in a "list processing" one. A single communication is only one item in a set related to one or more tasks. Each task results in a list of relevant communications [46].
- The future of CMC systems lies in the degree of tailoring that is possible within a single interface metaphor and the degree of integration with other computer resources [46].

For example, one of the standard design exercises this author gives students is the design of a "group calendar." It gives students a difficult problem because it does not take them very long to realize that the issues of communication structure, and the control of communication structure within the group calendar are closely tied to organizational culture and norms. Under what conditions and who has access to what information? One cannot design a single system that would be acceptable across a wide range of organizations.

3.0 Asynchronous Group Operations.

Much of the current work in the area of groupware and GDSS is carried out under the chronic fallacious assumption that has always plagued the development of information systems: the presumption of automation. It is assumed that the best way to do something is the way it was originally done manually. Although that may be the easiest thing to sell, it has been demonstrated consistently that this is among the worst ways to design a system to gain the benefits that computerization can offer.

The most misunderstood concept in CMC systems is the view that an asynchronous (or nonsimultaneous) communication process is a problem, because it is not the sequential process that people use in the face-to-face mode. The approach of "How do we make CMC feel to the user like face-to-face processes?" is incorrect. The real issue is how do we use the "opportunity of asynchronous communications" to create a group process that is actually better than face-to-face group communications?

The primary advantage of using CMC to support group processes is not that people can engage in the process whenever it is convenient for them. It lies in the very fundamental asynchronous nature of the communication medium. Many of the current design philosophies are trying to maintain the sequential nature of the process that groups go through in face-to-face settings, and assuming this is the right way to go. Quite the contrary, the potential for real improvement in the group processes lies in the fact that individuals can deal with that part of the problem they can contribute to at a given time, regardless of where the other individuals are in the process.

The advantages of groups are that individuals with very different psychology, expertise, and resulting approaches to problem solving contribute their "differences" to the group problem. It is well known that different people approach complex problems from very different directions. The potential for the computer is in the ability to allow people to do this, and to integrate the results for the group as a whole. The recognition of this was at the core of most good Delphi designs, and was one of the reasons that the paper-and-pencil communication structures of Delphi produced many underlying insights that would not be expected to occur in face-to-face processes for the same ends.

Figure 1 is a collection of tables from many different sources [7, 10, 17, 18, 27, 30, 35, 36, 49, 52] of the factors, processes, and dimensions that can be involved in both individual and group problem-solving activity. Most real situations may involve only a much smaller portion of this morphology. For individuals, there is the natural tendency to approach a problem with those cognitive processes at which they excel. For groups, the problem, and the nature of the group and its leadership, dictates some selection of an appropriate sequence of problem solving.

The specification of a group process may very well be in conflict with the ways in which specific individuals can best contribute to dealing with the problem. The resulting opportunity for asynchronous approaches to group problem solving is to free the individual to deal with the problem in ways consistent with his or her cognitive style. The resulting challenge for design is the communication structures and facilities to allow for synchronization of the group process, and the organization of the material for the benefit of the group.

Another important observation is that, for each and every group or individual process, one can hypothesize possible computer-based tools and facilities

Figure 1.
Asynchronous Group Process Factors

1.1 Group Problem Solving

Process	*Clarification*
Orientation	
Recognition	of an issue
Exploration	of related considerations
Formulation	of a strategy or approach
Generation	of specific ideas
Evaluation	
Focusing	on an item
Specification	of an item
Introspection	evaluating an item
Review	considering the results
Solution	
Closure	completion of an item
Presentation	finalizing expression of results

1.2 Individual Problem Solving

Process	*Clarification*
Orientation	
Identification	of an issue
Acquisition	of a consideration
Reformulation	of an item
Segmentation	Reduction and dividing
Exploration	
Abstraction	developing a generality
Search	finding related items
Structuring	considering relationships
Interpretation	gaining an understanding
Conceptualization	arriving at a concept
Evaluation	
Induction	empirical extrapolation
Deduction	logical relationships
Analogical reasoning	by analogy
Creation	leaps of inference
Selection	making decisions or choices
Scaling	relative evaluation of items
Introspection	consistency evaluation

1.3 Meta Individual and Group Processes

Process	*Clarification*
Regulation	
Sequencing	of tasks
Iteration	to prior tasks
Synchronization	of group process
Participation	by group members
Tracking	of status of process
Assignment	of tasks

1.3 Cont.

Process	*Clarification*
Facilitation	
Organization	classification of material
Abstraction	developing generalities
Summarization	coding and abbreviation
Filtering	eliminating noise
Exposure	of hidden factors
Retrieving	of related material
Integration	modeling
Social-Emotional	
Socialization	Cooperation and friendship
Signaling & Cueing	of status
Consensus Formulation	of group results
Conflict Exploration	reasons for disagreements
Value Exposure	Interest conflict exploration
Conflict Resolution	resolving disagreements

1.4 Objects of Discourse

Object	*Clarification*
Problems, Issues, Questions	main concerns
Goals, Objectives, Plans	Normative formulations
Strategies, Policies, Agendas, Approaches	Solution management
Concerns, Criteria, Arguments, Assumptions, Viewpoints, Opinions, Values, Interests	Underlying factors
Consequences, Scenarios, Impacts	Evaluative factors
Tradeoffs, Compromises, Proposals, Solutions, Allocations, Decisions, Projects, Tasks	Possibilities

1.5 Dimensions of Human Communication

Factor	*From*	*To*
Cooperation	Cooperative, Friendly	Competitive, Hostile
Intensity	Intense, Engrossed	Superficial, Uninvolved
Dominance	Democratic, Equal	Autocratic, Unequal
Formality	Personal, Informal	Impersonal, Formal
Orientation	Productive, Task Oriented	Unproductive, No Objective

1.6 Other Intervening Factors

Factor	*Clarification*
Norms	Of group and organization
Values	Of individuals
Pressures	Status, biases, hidden agendas
Task type	The nature of the problem
Technology	Interface and facilities
Leadership	Type and quality
Density	Size, relationships
Group type	Cooperative, Negotiating, etc.

1.7 Outcome Variables

Variable	*Clarification*
Consensus	Degree of group agreement
Agreement	Match between individual view and group view
Quality	Goodness of results
Confidence	In the results
Comprehension	Of the results
Efficiency	Relative effort in arriving at result
Commitment	of members to group results
Satisfaction	with group process
	with technology
	with own performance
	with other individuals
	with group results
	with leadership

that might be an aid in that step of the process. Many of these are very dependent on the nature of the problem. Hence, we are driven to the concept of a "toolbox" metaphor for group support. The toolbox should be tailorable by the individual in his or her individual examination of a problem, and by those holding appropriate roles in the group dealing with the problem.

4.0. EIES2, TEIES, and PERSONAL TEIES

For the past three years at NJIT, we have been working on what we feel is a new generation of technology for CMC systems. Our design is based, in large part, on the experiences of the past two decades and the assumptions and goals we have formulated as a result of these experiences. The basic CMC metaphor we are utilizing is summarized in Figure 2.

Many of these objects should be clear to anyone who has had some experience with this technology. Our discussion will focus on those areas where there are new specifics in the functionality of a given object. Two of the concepts that are very new, not explicitly present in current CMC systems, are "notifications" and "activities." These will be discussed in some detail. The final discussion will be of some of the underlying technical factors that make the functionality we are incorporating feasible.

4.1. Conferences

The heart of the working space for a group is a conference or a set of conferences. There may be many different conferences for a group, each serving a slightly different subobjective. Some of the classical types of conferences that require unique forms of tailoring are listed next.

Figure 2.
CMC Metaphor

Component	*Explanation*
Members	The users of the system
Personal Index	To track items
Roles	Privilege collections
Tickets	Privilege exceptions
Groups	Super members of the system
Membership List	Status of all members
Group Index	Shared by all members
Group Conference	Owned by group
Group Messages	Sent and received to group
Messages	Private, Group, and Public
Conferences	Topic communication space
Membership List	Status of all members
Conference Index	Comment keys
Comments	Entries in a conference
Directory	Of members, groups & conferences
Interest Index	Member interests & message sending key
Topic Index	Group and conference topics
Activities	Executable programs
Notifications	Transaction notices
Forms	Structured data collection
Attachments	File attachments
Lists	Collections of items (e.g. marked list)
Index Entries	
Reference Keys	pointers to items
Filters	Screening terms
Labels	Substitutions for commands or strings

- *General discussion:* Conferences structure to facilitate active participation by all members of a group or subgroup.
- *Tracking conferences:* Oriented to tracking events such as the status of a project, bug reports, etc.
- *Information exchange:* Oriented to organizing and categorizing information based on unpredictable inquiries and responses.
- *Planning conferences:* Oriented to structuring subjective viewpoints on complex situations.
- *Collaborative composition:* Oriented to structuring the creation of a document by a group of members.
- *Data collection and validation:* Organized to gather structured data, check the correctness of the data, and discuss the implications.
- *Games:* Use of role playing and event-oriented simulations for learning and exploring complex situations.
- *Training and education:* Use of Virtual Classroom™ structures for communications in a learning situation.

A conference, as a working space to meet a specific objective, can be structured in a number of ways. The resulting structure influences the efficiency of the conference communication facilities for reaching the specific conference objectives. The typical types of parameters that one can adjust and tailor when setting up a particular conference for a specific objective are the following:

- The memberships and roles of individual members are established and modified by the owner.
- Conference comments usually have a number of association and/or reply levels. For example, comment 24.5 would be the fifth reply to root comment 24. The owner of a conference should be able to specify the number of levels allowed and the size of the comments at each level. In a composition conference, fairly large comments are allowed. In a discussion conference, one wants to limit size to a few screens, because long comments break up the cognitive flow of the comments.
- In a composition conference, one should be able to restrict roles to fixed ranges of root comments. This way, one can assign selective composition and review responsibilities for different sections of a document.
- Determining the nature of the specific "activities" that occur in given conferences, when they should be activated, and who may activate them.
- The introduction of specific forms, to allow the gathering of structured information from members of the group and controlling the organization of the resulting data base.
- Specifying the degree of control on the incorporation of index keys.

The major mechanism for selective tailoring of the conference structure is the introduction of activities. By choosing the types of activities and when they are used, the owner or organizer of a conference has considerable flexibility, both in assigning what has to be done and in providing the ability of the group to track the progress of the effort.

4.2. Comment and Message Structure

The core of CMC systems is the communication objects that carry actual content between the members of a group or team. These are messages and conference comments. There are four distinct parts that can exist for a given message or comment. These are:

- *Abstract:* The abstract is all the status information associated with a message or comment. It includes:
  —Name and identification of author and any modifier, and date and time item was created or modified.
  —Identification of the item, size of item, number of replies, and pointers to related items.
  —Title of the item, key words for indexing of item, and status with respect to attachments and/or activities.

- *Content:* This is the actual content of the communication item. In most CMC systems, members are able to designate whether they want to see the content after reviewing the abstract of an item.
- *Attachments:* Attachments or appendages serve a wide variety of functions. They may represent an attached graphic or binary file produced by a special software package. The types of attachments that are possible are listed next.
  —Any type of file that is to be transferred to the member's workstation or personal computer before they can be viewed or utilized.
  —Very long text items that the sender describes in the contents, in order to give the receiver the opportunity to decide whether to display the longer attachment.
  —Appendages to the original text item made by various readers: expressions of reactions (such as agreement, disagreement, or approval), with the content of the item. Such expressions are identified by who in the group added the appendage. Canned notifications may be attached as well.
- *Activities:* These are described subsequently in detail. They are executable procedures that aid in the structuring of the activities of a group, or which facilitate the ability of individuals to handle large amounts of information. While one may "view" a comment or message, one would "do" the linked activity. For activities, the content of the comment or message would usually provide information about the activity.

Clearly, a great deal of the adaptability of structures in a CMC environment rests with the ability to use a text item in a significant number of different ways. Associated with this variety of utilization is the concept of notifications. For example, when someone modifies another person's item, the original author should be notified by the system of the occurrence.

4.3. Activities

One of the necessities for a CMC system to service group-oriented objectives is the integration of other computer resources within the CMC environment. This means the following types of available capabilities:

- Decision support tools collect and process and display "votes," such as weighting, ranking, or yes-no "straw votes" on options.
- Members of a CMC system should be able to bring data from other data bases into the communication environment.
- One must be able to trigger the execution of programs that support the group process and obtain the results of those programs within the conferencing environment.

The future direction for group support in the computer environment is toward the integration of communications and all other computer resources into a single interface structure. There is no reason why the same interface process utilized to communicate with other humans cannot be utilized to communicate with other computer- or network-based facilities. A communication system is the ideal concept for a user-oriented executive system.

There are many approaches to integration with the underlying technology. We utilize the metaphor of an "activity" that can be attached to any communication item, such as a comment or message. This activity, when triggered or done, will execute a program or procedure on the host computer or the network of computers. Such an activity could ask the person using it to supply input data, and generate a new communication item to receive the output of the resulting program.

As long as the preceding is implemented as a general facility, it can be used to easily trigger requests and searches of data bases, execute analysis programs, or make changes to data bases. The original communication item containing the attached activity can provide a description of the activity or a summary of its current status.

LIST GATHERING ACTIVITY

An example of a necessary activity is the ability to collect collaboratively a list of structured items and to treat this collection of items as one type of list. One list is the table of contents of a document, such as the document the group is trying to create. Another type of list is a set of tasks to be accomplished. In addition, there could be lists of issues to address, terms to define, alternative criteria for a decision, possible solutions to a problem, etc.

How to handle the contributions to a list varies with the nature of the objective:

- Any member can add an item to the list at any time, such as a list of issues to address.
- Only one, or a certain select set of individuals, can contribute to the list, as in a table of contents or a set of tasks to be done.
- Some individuals can add to only certain parts of the list, as in adding subheadings to certain sections of a table of contents.
- Individuals can be assigned to various items in the list, or they can volunteer to be associated with various items in the list. For example, who is to write a particular section of a document? Or, who is to handle a certain task?
- A status can be associated with items on the list, such as whether a task has been done and the date completed.

The person creating an activity has to have control over when certain actions are allowed. For example, if people are contributing to the list, the owner of the activity should be able to do things such as "close" it to further contributions, if it has sufficient contributions. The owner should be able to open, at the right time, certain other actions that can take place on an existing list:

- Associate a voting scale with items on the list, which may be of a number of different types. For example, a simple yes/no vote or an arbitrary 1–7 scale, which can be used to measure consensus on such dimensions as agreement, feasibility, importance, etc.
- Allow other text items to be associated with the items on the list, such as in the actual drafting of entries for the table of contents.
- Freeze contributions at any time, and perform editing and reorganization of the list.
- Allow selections of responsibility for items on the list, such as in a voluntary task assignment.
- Control whether a person has access to the activity before they have made a contribution to it.
- Open the activity to access by others who are not part of the original group developing the contents of the activity.

There are many types of activities possible, both of a very general and very specific nature. Whatever is the nature of a specific activity, it is important that this facility provide the tracking of status of activities for both individuals and the group. The system must provide individuals' organized reviews of what activities they have and have not done. The leaders or facilitators of a group need to know the status of who in the group has or has not completed a given activity. Also, they should have facilities for such things as sending automatic reminder notifications to anyone who has not done a particular activity.

In some types of tasks, it is important, for psychological motivation, that all the group members are able to perceive who in the group has and has not contributed to the group-oriented activities.

The concept of activities, within a CMC system, is open-ended and represents one of the primary mechanisms whereby future extensions will be made to EIES2 and TEIES. What is crucial is that the incorporation of a wide range of tailored facilities to support an application be provided via a common interface metaphor, with the same command functions and object definitions applying to the whole set. This is the only way that all the members of the group can quickly acquire and learn the tools as they are carrying out the application.

Once the members learn the commands (Figure 3) that apply to any and all activities, they are able to manipulate any type of activity. The related concept of notifications provides a single facility through which the progress of any type of activity can be tracked by the members of the group.

4.4. Notifications

Notifications are short one- or two-line messages intended to reduce the amount of effort necessary to communicate about complex tasks. These objectives are listed next.

Figure 3.
Activity Commands

Command	*Explanation*
View	Content of a message or comment Would usually explain nature of activity
Review	Displays status of activity (e.g., Done or Undone)
Do	To execute the activity (e.g., vote, read document, etc.)
Organize	Control the activity
Notify	Trigger associated notifications

- Generate automatic alerts based on transactions that have taken place. For example, a reply to an "urgent" item might notify the readership of that item that a reply exists. Most automatic alerts are optional in nature.
- Provide a direct manipulation handle, since one may point to a notification and retrieve the material to which it is referring.
- Provide a data base to the member for tracking the actions associated with a particular task. Each member has a cyclic file of the notifications delivered to him or her.
- Providing the interface function of "closure" on processes that the member or group have triggered. Each activity defined for the system may incorporate its own set of tailored notifications and the conditions under which they are triggered.
- Alerting members to changes of status, such as a vote summary being viewable.
- Reminders such as the controller of an activity triggering automatic sending of reminders to those who have not made their contribution.
- Reducing the need for messages or comments by the use of "canned" notifications. A member may point to an item and trigger a standardized notification expressing any of a large number of common communication tasks, such as expressing agreement or disagreement, desirable or undesirable, responsibility for taking care of, etc.

These notifications may be incorporated as attachments to the original message or comment.

- Enhance communication awareness though notifications, such as announcements about new members being added to a private conference or the introduction of new classification categories for use by the group.

As an example, if an individual sends a message that is a "planned decision," the receiver can respond with a canned notification by merely pointing to the message and triggering the "notify" command. This provides a menu of canned notifications, among which are the following:

AGREE (I agree with your decision.

DELAY (Wait until I respond before taking the action.)

Notifications represent a general open-ended concept and provide a great deal of flexibility as to how they may be used in association with groups, conferences, and activities.

The importance of the concept of notifications is that a wide variety of applications for alerting, closure, and tracking is served by one interface metaphor, and not by a variety of interfaces for different applications.

4.5. Roles, Privileges, and Tickets

One aspect that makes a CMC system more difficult to design and develop is that the system must incorporate the software to support the roles that humans take on in both facilitating and leading groups. In synchronous systems, this aspect is largely ignored.

Roles on our systems are built out of a subset of the primitive privileges that we have defined as being crucial to the human communication process. These are operations that allow people to do things such as adding something to someone else's file folder without being able to see what is there. One can do this in the physical world, and clearly systems that are going to offer the same flexibility must go far beyond the basic privileges of read and write. Based on empirical observation of human communications, we have defined some 25 such privileges. Some of these are also associated with maintaining a high degree of privacy and security with respect to the communication process. Some examples of such privileges are listed next.

- APPEND: which only allows material to be added to such objects as a conference comment, or message, without any necessary access to what is there unless one also has a READ privilege. It does not allow any modification of what is there, unless a separate MODIFY privilege is part of the privilege set.
- LINK: to be allowed to link someone else's object to another object.
- ASSIGN: to be able to establish a member for an activity such that the activity becomes required for the member.
- USE: allows user's machine to use data and/or code in a different user machine but does not allow transfer.

Clearly, users do not normally deal with individual privileges. Rather, they deal with roles. Some examples of the sort of roles that can be created out of the basic privileges are listed next.

- Indexer: the person in a conference or group who can modify and update the index, including the changing of keys on associated communication items.
- Organizer: the person who can create activities in a conference.
- Contributor: a person who can add comments to a conference, or send group messages, but cannot see any of the others that have been created.

The final component of this facility is the concept of TICKETS. A ticket is the ability of a user to pass a specific privilege that they possess to another user. For example, one could give a secretary a privilege to edit an authored comment in a conference, even though the secretary has no access to the rest of the conference. Tickets can be made conditional on such things as the number of times they can be used and the interval of time during which they may be used. A ticket can be canceled by the issuer at any time. Usually, tickets will generate notifications back to the issuer when they are used. They can also be designated as transferable, if the situation warrants it. Tickets are a mechanism to handle the unpredictable needs for the usage of privileges.

5.0. APPROPRIATE TECHNOLOGY FOR CMC

For the past three years, we have been developing a new generation of CMC systems. EIES2 is now available for installation at other sites. It is a fully distributed CMC system that operates under UNIX in any TCP/IP or X.400 network, with distributed "group" and "user" agents [50]. The IBM/VM system called TEIES (Tailorable EIES [46]) is currently in alpha test. Internally, both systems utilize a very similar design philosophies, based on what we have come to feel are natural consequences of the requirements for group support in a communications environment.

There are a large number of factors that lead one quite naturally to the necessity of an object-oriented data base as the foundation for any CMC system. These are as follows:

- Members and groups are represented in the data base and must be linked to the communication objects with which they have some association.
- Maintaining the security of communications along with the flexibility of a wide range of alternative communication-oriented privileges of access,
- Minimizing user-interface learning by the use of generic commands that may have somewhat different functionality, as a result of the type of object on which they are used.
- In any collaborative environment, communicators may make use of communications produced by others. Creators of items must be made aware that their items are being used by others if they seek to modify such items.
- The need to integrate other computer and information resources with the CMC system in such a way as not to violate security and privacy requirements.

The security and privacy of communications must be maintained rigorously in a communication system that supports organizational communications and the activities of various intersecting and overlapping groups. A second fundamental requirement is that the system must allow the same sort of communication flexibility humans have available with other communication media. A third is the necessity to integrate other computer resources that the individuals and groups need to support dealing with their tasks and problems.

The preceding considerations lead to a number of important technical requirements that form the basis for our current systems:

- The data base must be self-contained with respect to privileges of access to communication objects by the users of the system.
- Members must be objects defined in the data base.
- Groups, which are collections of members, must also exist as objects, so that linkages between communication objects may be made to groups as well as to members.
- The essence of a link is the specification of a bit mask that indicates the set of primitive privileges associated with the connection between a communications object and the member or specific groups to which he or she belongs.
- Each object type must have its own set of associated functions, so that generic user commands can be interpreted in terms of the object on which they are working.
- Links between objects must be two-way, so that actions on one object can be transmitted, as appropriate, to other linked objects. As a result, an object-oriented data base is a very natural approach to handling many of the requirements for CMC systems. In addition, object-oriented systems allow the behavior of an object to be defined independently from the rest of the system. This allows for the efficient evolution of the system through the introduction of new object types. Both EIES2 and TEIES have been designed around the basic object-oriented data-base concept.

With the addition of privileges within the data base, it is possible for the data base alone to determine whether or not to honor a request for data from any other system or application program. The EIES2 data base can be accessed from C programs or the UNIX shell. The TEIES data base can be accessed utilizing REXX. This means that other application programs can be created that make use of the CMC data base, without the application programmers having to be concerned with any aspect of security or privacy. In addition, any such application program can be triggered through the use of an "activity," within the CMC interface itself. This provides for the integration of other computer resources, either through an internal (e.g., activities) or external interface.

Both TEIES and EIES2 are distributed systems. EIES2 is designed around the ISO model and is composed of separate user and group agents, which may be located on separate machines. Each agent has its own Remote Operation Server, based on the X.400 standards. There is a Communication Language Processor, which accepts X.409 language specifications to create new objects for the data base. In EIES2, a single conference has a master copy on a single group agent, but the distributed nature of the system is completely transparent to a user. In principle, EIES2 can support a very large user population, given a sufficient number of machines within a TCP/IP or X.400 network. At the moment, we have three HP machines and a SUN machine at NJIT, and an HP machine in Denmark, all operating as one system.

TEIES is designed to be distributed as a single master virtual machine, with multiple user and data-base machines. The master machine provides a directory service for all objects in the system, and it also validates the establishment of a link between a given user machine and a given data-base machine with a specific set of access privileges. TEIES is designed to be very compatible with the IBM/VM environment. REXX can be used as the integration language.

Given the perspective of the evolution of a CMC system within an organizational context, the next obvious requirement is to minimize the effort involved in creating new functionality and interface capabilities. EIES2 utilizes a fully distributed SMALLTALK interpreter, developed at NJIT to create both the interface and user functionality. The applications programmer does not have to be concerned about the location of objects in the network of user and group agents. The TEIES interface is written in a version of SGML (Standard General Markup Language) developed at NJIT, which has implicit knowledge of the TEIES objects through the defined object and data variable names in the TEIES data base. It is our view that SGML is ideal as an interface specification language, given the proper functional extensions. It allows the interface specification to be stored in the text items that are a part of the CMC system. This also means that different users can be using interfaces in different languages on the same operational version of the system.

The final component of the NJIT development effort in CMC systems is PERSONAL TEIES™. This MS/DOS-based system serves as a graphics composition and display system, as well as a communications front end to TEIES and EIES2. One may view PERSONAL TEIES as, in part, a GKS to NAPLPS translator. GKS is used to drive the workstation, and NAPLS is the communication format for the graphics, as well as the data storage specification. PERSONAL TEIES™ is designed to allow a group to exchange and modify conceptual diagrams. It is intended to provide a form of "group graphics," where members in the group may make modifications to each other's diagrams. An important functionality of PERSONAL TEIES™ is that users may create their own icon sets. These could be anything from math symbols to symbolic representations of parts of an organization or project. Therefore, a group can develop its own diagrammatic jargon for representing graphical relationships of interest. As a result, many situations (e.g., laying out a new manufacturing process using a set of machines) can be reduced to the process of layouts and icons developed by the group. We plan to transfer PERSONAL TEIES™ to the Macintosh environment in the future.

All our software systems are being developed with the C language as the primary foundation language. As one might guess, there is a strong emphasis on utilizing standards wherever possible. These systems have been designed to be both usable production systems and provide a foundation for research and development in CMC and the related areas of Groupware, GDSS, and CSCW. Furthermore, the facilities have been provided to make it easy for organizations to incorporate their own tailoring and integration with other information systems and resources. We expect their evolution to stretch over the next decade, and to see other organizations making their own extensions to the basic foundation.

6.0. SUMMARY

The asynchronous group support problem is an order of magnitude more difficult than working with synchronous groups. The system must be able to aid in the organization of the material over long periods of time. It must also provide support for human roles and communication signaling and cueing. However, in the asynchronous approach, it is possible to integrate individual problem-solving processes with group problem-solving processes in such a manner that one can hope for true "collective intelligence" by taking advantage of both processes. Systems need to be designed based on appropriate models and theories of both processes.

The role of a well-designed CMC system is to support the human roles dedicated to synchronizing the individual and group processes, and to integrate the toolbox needed to support dealing with complex problems. For example, a typical planning effort for a new product or production process may involve the specification and analysis of anywhere from a few hundred to a few thousand subprojects and tasks. This has to be accomplished by a multitude of individuals and groups with overlapping and shifting membership. Such efforts need the benefits of support tools in such areas as structural modeling that will deal with collaborative and conflicting estimates of costs, effort, time to accomplish, relative significance, alternatives, objectives, etc.

There is no reason why a CMC system cannot support synchronous as well as asynchronous use. The focus on only synchronous systems is very much the classical "Drunkard's Paradox" of providing an easy solution to the wrong problem. A great deal of the current synchronous work seems to be oriented at building the "telescope." There is a tremendous social technology of small group problem solving, and it is not clear that the claimed effectiveness of elaborate and expensive decision rooms outperforms a well-structured "focus" group with normal meeting room facilities. At least, this sort of controlled experiment is not common in the literature.

The real advantages of computer technology lie in the integration into the group process of the powerful analysis techniques that can aid the solution of complex problems. These include:

- Scaling methods, such as Multi-Dimensional Scaling, to aid in the understanding and organization of human judgments [19, 27, 37].

- Structural modeling methods, such as Interpretive Structural Modeling and Cross Impact Analysis [25, 27], to aid in dealing with large-scale complex problems.
- Modeling, simulation, and gaming methods [20, 27] to aid in the forecasting of long-term implications.

The incorporation of techniques in these areas is what will bring about, in the long run, sizable benefits for group processes. However, they all require learning curves and facilitation in interpretation that mean availability of these aids on a continuous asynchronous basis. Complex group problems, such as "parking cars," already exist, and the need is to focus on those problems.

Furthermore, individuals will not tolerate, in the long run, a host of different systems for dealing with the same tasks. They will also not tolerate being constrained to one sequential approach for solving all problems. CMC systems have to be viewed as "toolboxes" that can be tailored by the users, both on an individual and group basis, for dealing with a specific type of problem. The ability of users to "self-tailor" the system and the integration of the tools a computer network can provide are the primary challenges for the designers of CMC systems.

To date, CMC systems have been designed and utilized to support real applications in such areas as project management, crisis management, planning and budgeting, collaborative learning, collaborative composition, group therapy and meditation, gaming, data-base validation, large-scale information exchange, decision support, and Delphi exercises. The key to good design is as much the understanding of the task as it is understanding the technology. Designers should be aware that they are not merely designing a computer system, they are designing a "social system" [40]. Perhaps even more important, management needs to be aware that this is what is occurring. These systems have and will continue to change the nature of management and the behavior of the organization by providing an evolutionary approach to the development of these systems by coupling an evaluation program [23].

7.0 REFERENCES

1. Bahgat, A. *A decision support system for zero-base capital budgeting: A case study*, Ph.D. in Management Thesis, Rutgers Graduate School of Management, Newark, N.J., 1986.
2. Bernstein, L.M., Siegel, E.R., and Goldstein, C.M. The hepatitis knowledge base: A prototype information transfer system. *Annals Internal Medicine 93*, 2 (1980), 169–181.
3. Cashman, P.M. A communication-oriented approach to structuring the software maintenance environment, ACM SIGSOFT. *Software Engineering Notes 5*, 1 (1980), 4–17.
4. Churchman, C.W. *The Design of Inquiry Systems.* Basic Books, New York, 1971.
5. Conklin, J., and Begeman, M. gIBIS: A Hypertext tool for team design deliberation. *ACM Trans. Office Info. Systs. 6*, 4, (1988), 303–331.
6. Daft, R.L., and Lengel, R.H. Organizational information requirements, media richness and structural design. *Manage. Sci. 32*, 5 (May 1986), 554–571.
7. Delbecq, A.L., VandeVen, A., and Gustofson, A. *Group Techniques for Program Planning: A Guide to Nominal Group Techniques and Delphi Processes.* Scott-Foresman, 1975.

8. Englebart, D.C. Coordinated information services for a discipline or mission oriented community. In *Proceedings of the 2nd Annual Computer Communications Conference.* SRI, San Jose, 1973.
9. Englebart, D.C., Watson, R.W., and Norton, J.C., The augmented knowledge workshop. In *Proceedings of the AFIPS National Conference 42,* 1976.
10. Ericsson, K.A., and Simon, H. *Protocol Analysis: Verbal Reports as Data.* MIT Press, Cambridge, Mass., 1984.
11. Hiltz, S.R. *Online Communities: A Case Study of the Office of the Future.* Ablex Press, Norwood, N.J. 1984.
12. Hiltz, S.R. The "Virtual Classroom": Using computer-mediated communication for university teaching. *J. Commun. 36,* 2 (Spring 1986), 95–104.
13. Hiltz, S.R., Johnson, K.J., and Turoff, M. Experiments in group decision making, 1: Communication process and outcome in face-to-face vs. computerized conferences. *Human Commun. Res. 13,* 2 (Winter 1986), 225–253.
14. Hiltz, S.R., and Turoff, M. Structuring computer-mediated communications to avoid information overload. *Commun. ACM 28,* 7 (July 1985), 680–689.
15. Hiltz, S.R., and Turoff, M. *The Network Nation: Human Communication via Computer.* Addison-Wesley, Reading, Mass., 1978.
16. Hiltz, S.R., Turoff, M., and Johnson, K. Using a computerized conference system as a laboratory tool. *SIGSOC Bull. 13,* 4 (1982), 5–9.
17. Hirokawa, R.Y., and Poole, M.S. *Communication and Group Decision-Making.* Sage Publications, Beverly Hills, Calif., 1986.
18. Hoffman, L.R. *The Group Problem Solving Process: A Valance Model.* Praeger Publishers, New York 1979.
19. Hopkins, R.H., Cambell, K.B., and Peterson, N.S. Representations of perceived relations among the properties and variables in a complex system. *IEEE Trans. Syst. Man, Cybern. 17,* 1 (Jan./Feb. 1987), 52–60.
20. Hsu, E. Role-event gaming-simulation in management education: A conceptual framework and review. *Simulation and Games 20,* 4 (Dec. 1989), 409–438.
21. Johnson-Lenz, P. and T., Kerr, E., and Hiltz, S.R. Groupware: The process and impacts of design choices. In *Computer Mediated Communication Systems,* E., Kerr and S.R. Hiltz, (Eds.). Academic Press, New York, 1982.
22. Johnson-Lenz, P. and T., and Hessman, J.F. JEDEC/EIES computer conferencing for standardization activities. In *Electronic Communication: Technology and Impacts,* M. Henderson (Ed.). AAAS Selected Symposium 52. Westview Press, Boulder, Colo. 1980.
23. Kerr, E., and Hiltz, S.R. *Computer-Mediated Communication Systems.* Academic Press, New York, 1982.
24. Kupperman, R.H., Wilcox, R.H., Smith, H.A. Crisis management: Some opportunities. *Science 187* (1975), 404–410.
25. Lendaris, G. Structural modeling: A tutorial guide. *IEEE Trans. Syst., Man, Cybern. 10,* 12 (Dec. 1980), 807–840.
26. Lerch, I.A. Electronic communications and collaboration: The emerging model for computer aided communications in science and medicine. *Telematics and Informatics 5,* 4 (1988), 397–414.
27. Linstone, H., and Turoff, M. *The Delphi Method: Techniques and Applications.* Addison-Wesley, Reading, Mass., 1975.
28. McKendree, J.D. Decision process in crisis management: Computers in a new role. In *Encyclopedia of Computer Science and Technology,* J. Belzer (Ed.), Volume 7. Marcel-Dekker, 1977.
29. Mitroff, I.I., Nelson, J., and Mason, R. On management myth—Information systems. *Manage. Sci. 21,* 4 (1974), 371–382.
30. Pinsonneault, A., and Kraemer, K. The impact of technical support on groups: An assessment of the empirical research. *Decis. Support Syst. 5* (1989), 197–216.
31. Renner, R., et al. EMISARI: A management information system designed to aid and involve people. In *Proceedings of the International Symposium on Computer and Information Science (COINS). Dec. 1972.*
32. Schneider, S.J., and Tooley, J. Self-help computer conference. *Computers and Biomedical Research 19* (1986), 274–281.

33. Schneider, S.J. Trial of an on-line behavioral smoking cessation program. *Computers in Human Behavior 2* (1986), 277–296.
34. Siegel, E.R. The use of computer conference to validate and update MLM's hepatitis data base. In *Electronic Communications: Technology and Impact*, H. Henderson and J. MacNaughton (Eds.), AAAS AAAS Selected Symposium 52. Westview Press, Boulder, Colo. 1980.
35. Sternberg, R.J. Human intelligence: The model is the message. *Science 230*, 4730 (Dec. 6, 1985), 1111–1118.
36. Sternberg, R.J. (Ed.). *Handbook of Human Intelligence*. Cambridge University Press, Cambridge, 1982.
37. Torgerson, W.S. *Theory and Methods of Scaling*. Wiley, 1958.
38. Turoff, M. Delphi and its potential impact on information systems. In *AFIPS Conference Proceedings 39*, (1971), 317–326.
39. Turoff, M. Delphi conferencing: Computer based conferencing with anonymity. *J. Technol. Forecasting and Social Change 3*, 2 (1972) 159–204.
40. Turoff, M. Management issues in human communication via computer. In *Emerging Office Systems*, R. Landau et al. (Eds.). Ablex Press, Norwood, N.J. 1982.
41. Turoff, M. Information and value: The internal information marketplace. *J. Technol. Forecasting and Social Change 27*, 4 (July 1985), 257–373.
42. Turoff, M. The anatomy of a technological innovation: Computer mediated communications. *J. Technol. Forecasting and Social Change 36*, (1989), 107–122.
43. Turoff, M., and Hiltz, S.R. Computer support for group versus individual decisions. *IEEE Trans. Commun. 30*, 1 (Jan. 1982), 82–90.
44. Turoff, M., and Hiltz, S.R. Office augmentation systems: The case for evolutionary design. In *Proceedings of the 15th Hawaii International Conference on Systems Science*. Jan. 1982, 737–749.
45. Turoff, M., and Hiltz, S.R. The electronic journal: A progress report. *J. ASIS 33*, 4 (July 1982), 195–202.
46. Turoff, M., and Hiltz, S.R., Foster, J.F., and Ng, K. Computer mediated communications and tailorability. In *Proceedings of the Twenty-Second Annual Hawaii Conference on Systems Science*. Vol. III. IEEE Computer Society, Jan. 1989, 403–411.
47. Turoff, M., Whitescarver, J., and Hiltz, S.R. The Human-machine interface in a computerized conferencing environment. In *Proceedings of the IEEE Systems, Man and Cybernetics Annual Conference. 1977, 145–157.*
48. Vallee, J., Johansen, R., Lipinski, H., and Wilson, T. Pragmatics and dynamics of computer conferencing: A summary of findings from the forum project. In *Proceedings of the International Conference on Computer Communications*. IEEE, 1976, 208–213.
49. Wang, C.H., and Hwang, S.L. The dynamic hierarchical model of problem solving. *IEEE Trans. Syst., Man, Cybern. 19*, 5 (Sept./Oct. 1989), 946–954.
50. Whitescarver, J., et al. A network environment for computer supported collaborative work. In *Proceedings of ACM SIGCOMM Workshop: Frontiers in Computer Communications*. Aug. 1987, 230–244.
51. Wilcox, R.H., and Kupperman, R., An on-line management system in a dynamic environment. In *Proceedings of the First International Conference on Computer Communications. IEEE*, 1972, 117–120.
52. Wish, M., and Kaplan, S.J. Toward an implicit theory of interpersonal communication. *Sociometry 40*, 3 (1980), 234–246.
53. Zinn, K.L., Parnes, R., and Hench, H. Computer-based educational communication at the University of Michigan. In *Proceedings of 31st ACM National Conference*. 1976.