INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

**A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600**

A GRAPHICAL, SELF-ORGANIZING APPROACH TO CLASSIFYING
ELECTRONIC MEETING OUTPUT

by

Richard Eldon Orwig

A Dissertation Submitted to the Committee on

BUSINESS ADMINISTRATION

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 9 5

UMI Number: 9603707

UMI Microform 9603707

Copyright 1995, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI

**300 North Zeeb Road
Ann Arbor, MI 48103**

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

2

As members of the Final Examination Committee, we certify that we have
read the dissertation prepared by RICHARD ELDON ORWIG
entitled A GRAPHICAL, SELF-ORGANIZING APPROACH TO CLASSIFYING
ELECTRONIC MEETING OUTPUT

and recommend that it be accepted as fulfilling the dissertation
requirement for the Degree of Doctor of Philosophy/Management

Hsinchun Chen
Dr. Hsinchun Chen, Advisor

12/19/94
Date

J. F. Nunamaker, Jr.
Dr. Jay F. Nunamaker, Jr.

12/19/94
Date

Douglas R. Vogel
Dr. Douglas Vogel

12/19/94
Date

Date

Date

Final approval and acceptance of this dissertation is contingent upon
the candidate's submission of the final copy of the dissertation to the
Graduate College.

I hereby certify that I have read this dissertation prepared under my
direction and recommend that it be accepted as fulfilling the dissertation
requirement.

Hsinchun Chen
Dissertation Director

5/10/95
Date

Dr. Hsinchun Chen

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED:

A handwritten signature in dark ink, appearing to read "Richard E. O'Leary", is written over a horizontal line.

Acknowledgements

This dissertation is a milestone to what I hope is a lifetime of continuous learning. My efforts to reach this milestone were made through much cooperation, support, and, sometimes, neglect of family, co-workers, and friends. This dissertation is more than a product of my efforts. It is also a product of the love and emotional support of my wife, Sally, and my daughters, Michaela and Rebecca. In addition to this love and support, I could not have made these efforts without sacrifice on Sally's behalf for the taking of more than her fair share of family obligations as well as financial support.

I must also acknowledge the patience and support of my advisor, Dr. Hsinchun Chen. He provided and continues to provide a model of academic excellence in research and teaching.

My financial support was provided through a research contract with the United States Army Construction Engineering Research Laboratory. Drs. Jay Nunamaker and Doug Vogel were instrumental in providing guidance and support for obtaining this contract as well as encouragement in developing facilitation skills. I am also indebted to Dr. Windell Ingram, Mark Bovelsky, Lynne Mikulich, the Defense Environmental Security Corporate Information Management office, and many others for their support. Even greater debts are owed to other project team members: Dr. Mark Pendergast, Andy Hartman, Doug Dean, Jim Lee, Jon Stapley, Boris Nevstrujev, Nina Katic, Jeff McKibben, Andrea Houston, and Ken Walsh.

This work is only part of a research stream directed by Dr. Chen. Other research project team members who directly or indirectly contributed to my work include Karen Ng, Joanne Martinez, Chris Schuffels, Dorbin Ng, Siunie Sutjaho, and Dr. Jerome Yen.

Dedication

To Sally, Michaela, Rebecca, my parents and family

TABLE OF CONTENTS

	<u>PAGE</u>
LIST OF ILLUSTRATIONS	Page 9
LIST OF TABLES	Page 10
ABSTRACT	Page 11
CHAPTER 1: INTRODUCTION	Page 13
CHAPTER 2: THE PROBLEM -- CLASSIFICATION IN EMS	Page 18
2.1 DESCRIPTION OF EMS	Page 20
2.1.1 Description of Hardware, Software, and Facilitation	Page 20
2.1.2 Examples of Types of Problems Addressed by EMS	Page 25
2.1.3 Typical Problem Solving Sequence	Page 27
2.1.3.1 Goal-Directed	Page 28
2.1.3.2 General Three-Stage Approach	Page 30
2.1.3.3 Tool Selection	Page 33
2.1.4 Strengths of EMS	Page 34
2.2 CONVERGENCE PROBLEMS IN A GROUP SETTING	Page 36
2.2.1 Difficulties with EMS Convergence	Page 37
2.2.1.1 Information Overload	Page 37
2.2.1.2 Vocabulary Problem	Page 38
2.2.1.3 Increased Cognitive Demand	Page 39
2.2.1.4 EMS Interface Problems	Page 40
2.2.2 An EBS/IO Example	Page 41
2.2.3 Techniques for Facilitating Group Idea Organization	Page 48
2.2.3.1 Avoid EBS Complexity Completely	Page 48
2.2.3.2 Divide and Conquer Method	Page 49
2.2.3.3 Limit the Input	Page 49
2.2.3.4 Keyword Option in EBS with Group Input	Page 51
2.2.3.5 Keyword in EBS with Facilitator/Leader	Page 51
2.2.4 Desired Classification: The Topic List	Page 52
2.2.5 Need for Automated Support	Page 55
2.2.5.1 Low Group Satisfaction	Page 56
2.2.5.2 Linking Comments to Topics	Page 58
2.2.5.3 Shortening Convergence Time	Page 59
2.2.5.4 Consistency in Topic Identification	Page 60
2.2.6 Desired Solution: Automatic Classification of EBS Comments	Page 60

CHAPTER 3: CLASSIFICATION LITERATURE REVIEW	Page 61
3.1 COGNITIVE VIEW	Page 61
3.1.1 Modeling Knowledge	Page 61
3.1.2 Summarization	Page 65
3.1.3 Abstraction	Page 68
3.2 SYSTEMS ENGINEERING VIEW	Page 69
3.2.1 Textual Analysis	Page 69
3.2.1.1 Automatic Indexing	Page 70
3.2.1.2 Natural Language Understanding	Page 74
3.2.2 Cluster Analysis	Page 78
3.2.2.1 Serial Approaches	Page 78
3.2.2.2 Parallel Approaches - Neural Network Clustering	Page 82
CHAPTER 4: RESEARCH DESIGN AND PREVIOUS WORK	Page 91
4.1 RESEARCH QUESTIONS	Page 93
4.2 HOW PEOPLE CLASSIFY EBS COMMENTS	Page 94
4.2.1 A Serial Comparison of Each Comment	Page 94
4.2.2 An Example: Human Approach to Real-time Classification ...	Page 97
4.3 HOPFIELD NET CLASSIFICATION - PREVIOUS WORK	Page 99
4.3.1 Automatic Indexing of EBS Comments	Page 100
4.3.2 Concept Space Generation	Page 102
4.3.2.1 Term Frequency and Document Frequency Computation	Page 102
4.3.2.2 Determining Threshold Through Information Loss Analysis	Page 102
4.3.2.3 Combining Weights	Page 103
4.3.2.4 Concept Space Generation	Page 103
4.3.3 Hopfield Net Classification	Page 105
4.3.3.1 Assigning Connection Weights	Page 106
4.3.3.2 Initialization with Unknown Input Pattern	Page 106
4.3.3.3 Activation and Iteration	Page 107
4.3.4 Hopfield Classification Example	Page 109
CHAPTER 5: KOHONEN CLASSIFICATION FOR EBS COMMENTS ..	Page 110
5.1 KOHONEN CLASSIFICATION ALGORITHM	Page 112
5.1.1 Initialize the Kohonen Layer	Page 112
5.1.2 Process Input	Page 113
5.1.3 Adjust Neighbors	Page 113
5.2 KOHONEN CLASSIFICATION EXAMPLE	Page 114
5.2.1 Convert Index and Frequency List	Page 114
5.2.2 Create Unit Vector Input	Page 115
5.2.3 Create the Map	Page 115

5.2.4 Interpret the Output	Page 118
CHAPTER 6: EVALUATION OF KOHONEN CLASSIFICATION	Page 121
6.1 EXPERIMENT DESCRIPTION	Page 121
6.1.1 First Stage: List Comparison	Page 122
6.1.2 Second Stage: Term Relevance	Page 124
6.1.3 Third Stage: Term Association	Page 126
6.2 SUBJECTS OF THE EXPERIMENT	Page 128
6.3 EXPERIMENTAL RESULTS	Page 128
6.3.1 Kohonen is Less Precise Than Facilitator	Page 130
6.3.2 Relevance is a Problem for Both Hopfield and Kohonen	Page 133
6.3.3 Kohonen Associates Terms as well as the Facilitator	Page 134
CHAPTER 7: CONCLUSIONS, CONTRIBUTIONS, AND FUTURE DIRECTION	Page 141
7.1 CONCLUSIONS	Page 141
7.2 CONTRIBUTIONS	Page 143
7.3 FUTURE DIRECTIONS	Page 144
APPENDICES	Page 147
A -- EXPERIMENT HANDOUT	Page 147
B -- PASCAL CODE: COMMENT CONVERSION TO KOHONEN INPUT FORMAT	Page 160
C -- PASCAL CODE: CREATE A SET OF UNIT VECTORS	Page 165
D -- PROCESS FOR RUNNING KOHONEN SOM ON EBS OUTPUT	Page 166
REFERENCES	Page 179

LIST OF ILLUSTRATIONS

Figure 1 -- Electronic Meeting System Facility	Page 21
Figure 2 -- EMS Software Support	Page 22
Figure 3 -- Electronic Meeting System Research Model	Page 24
Figure 4 -- Decomposition of a Complex Group Problem	Page 29
Figure 5 -- GroupSystems Electronic Brainstorming Participant Screen	Page 42
Figure 6 -- Sample EBS Output	Page 44
Figure 7 -- GroupSystems Idea Organizer Beginning Screen	Page 45
Figure 8 -- GroupSystems Idea Organizer Add Entry Screen	Page 46
Figure 9 -- Sample Solution to Group Classification Problem	Page 53
Figure 10 -- Top-Down and Bottom-Up Process	Page 55
Figure 11 -- Satisfaction level as a function of tool usage	Page 57
Figure 12 -- A process for Systems Development Research	Page 91
Figure 13 -- Cluster Analysis Output (Chen, 1994)	Page 93
Figure 14 -- Facilitator-Generated Topic List	Page 99
Figure 15 -- Sample Automatic Indexing Output	Page 101
Figure 16 -- Terms Ranked according to decreasing document frequency	Page 104
Figure 17 -- Hopfield Architecture and Algorithm	Page 108
Figure 18 -- Hopfield Classifier Generated List	Page 109
Figure 19 -- Kohonen Self-Organizing Map Architecture	Page 110
Figure 20 -- Kohonen Output	Page 119
Figure 21 -- Indexing output examples	Page 167
Figure 22 -- Terms List in Decreasing Order of Frequency	Page 168
Figure 23 -- Sample Kohonen Input Data	Page 169
Figure 24 -- Neighborhood Arrangements	Page 170
Figure 25 -- Assignment of Comments to Grid	Page 173
Figure 26 -- Unit Vector Input for Mapping Terms to the Grid	Page 174
Figure 27 -- Grid Assignment of Terms to the Map	Page 175
Figure 28 -- Kohonen Self-Organizing Map -- 10 x 20 grid	Page 176

LIST OF TABLES

Table I -- Three Concept Lists	Page 123
Table II -- Pooled List of Identified Terms	Page 125
Table III -- List of Association Terms	Page 127
Table IV -- Stage One Results	Page 131
Table V -- RECALL Statistical Analysis	Page 133
Table VI -- PRECISION Statistical Analysis	Page 134
Table VII -- Relevance Statistical Analysis	Page 135
Table VIII -- Associated Pairs for Each List	Page 136
Table IX -- Term Association Statistical Analysis	Page 137
Table X -- Association of Identified Pairs of Terms	Page 138
Table XI -- Recall/Precision Statistical Analysis of Associated Pairs	Page 140

ABSTRACT

This dissertation describes research in the application and evaluation of a Kohonen Self-Organizing Map (SOM) to the problem of classification of Electronic Brainstorming output. Electronic Brainstorming is one of the most productive tools in the Electronic Meeting System called GroupSystems. A major step in group problem solving involves the classification of Electronic Brainstorming output into a manageable list of concepts, topics, or issues that can be further evaluated by the group. This step is problematic due to the information overload and cognitive load of the large quantity of data.

This research builds upon previous work in automating the classification process using a Hopfield Neural Network. Evaluation of the Kohonen output in comparison with the Hopfield and human expert output over the same set of data found that the Kohonen SOM performed as well as a human expert in the recollection of associated term pairs and outperformed the Hopfield Neural Network algorithm. Using information retrieval measures, recall of concepts using the Kohonen algorithm was equivalent to the human expert. However, precision was poor.

The graphical representation of textual data produced by the Kohonen SOM suggests many opportunities for improving information management of textual electronic information. Increasing uses of electronic mail, computer-based bulletin

board systems, and world-wide web textual data suggest an overwhelming amount of textual information to manage. This research suggests that the Kohonen SOM may be used to automatically create "a picture that can represent a thousand (or more) words."

CHAPTER 1: INTRODUCTION

This research concerns capturing intelligence in a group problem-solving situation. Even before John McCarthy coined the term "artificial intelligence," AI, in the 50's, a comprehensive meaning of intelligence has been in debate. This philosophical debate continues today. However, most would agree that the ability to solve complex problems is one mark of intelligence.

One aspect of problem solving is an ability to decompose a problem. Humans depend upon their ability to decompose each problem into component parts and to understand the relationships among those components in order to understand the problem. These relationships include static relationships such as the hierarchical organization of problem components as well as dynamic relationships such as explaining how components interact in certain conditions. It is assumed that experts in a given domain have better understandings of the domain objects and relationships than the average person. In many cases, this level of understanding is sufficient for an expert to observe a given problem, comprehend why the problem exists, and prescribe a solution. "Knowledge" sometimes connotes this static state of understanding. However, complex problems may require further testing or rearrangement of the components by the problem solver in order to better understand the problem. "Innovation" or "creativity" is sometimes attributed to knowing how to rearrange, add, or delete the components of a problem (thus changing some relationships) in new ways

that more efficiently address a problem solution. The problem solver may knowingly manipulate components of a complex problem to better understand the particular problem and learn more about the components, their relationships, and, potentially, suggestions for problem solutions. For the purpose of this paper, intelligence is meant to imply both the knowledge and creativity necessary to solve a given problem.

To learn is to add to our base of knowledge. Using the definition of intelligence above, it logically follows that one method of "learning" occurs through the addition to the state of knowledge that results from objective observation of the manipulation of the objects and their changing relationships. Indeed, the foundation of the scientific method in gaining knowledge rests on the definition of "objective" in this "objective observation" (Strauss & Corbin, 1990) (Chisholm, 1966)(Salmon, 1967) (Boulding, 1956).

Elaine Rich defines artificial intelligence to be "the study of how to make computers do things which, at the moment, people do better (Rich & Knight, 1991)." If we adopt this definition of AI, an artificial intelligence approach to solving a particular human problem may result in adding to our base of knowledge in that domain -- particularly if we, as humans, do not completely understand how we solve the problem. For example, chess masters may not initially be able to explain their expertise. However, the process of trying to make computers Grand Masters in chess

adds to our understanding of how humans reason. In the broadest sense, software engineering is a learning process in that engineers must break down a human process into components (algorithms and data structures) in order to help solve a human problem. The software program is a computer model of a real-world problem solution. Again, in a broad sense, building this model is "knowledge engineering." Knowledge acquisition in building expert systems is the specific practical area of AI that transfers human knowledge into computer form. Knowledge acquisition can thus refer to both capturing expert knowledge in computer form as well as adding to human knowledge in better understanding the components, relationships, and processes associated with solving a problem.

Thus, an artificial intelligence approach to solving a problem that currently exists for humans may add to a base of knowledge if: (1) new understandings of objects and relationships occur as a result of capturing the human problem-solving process in computer hardware and software or (2) new understandings of problems and solutions occur as a result of observing the model of the problem that exists in its computerized form.

This research uses an artificial intelligence approach to understanding the problem of classification of concepts in an electronic meeting setting. Electronic meeting systems (EMSs) provide support for large groups to interactively work on a single

problem or collection of problems (Nunamaker, Dennis, Valacich, Vogel, & George, 1991) (Vogel, Nunamaker, Martz, Grohowski, & McGoff, 1989). Large groups of people use a network of computers to discuss complex organizational problems electronically. These electronic discussions create large quantities of text in a very short period of time. A major stage in the group problem solving process involves classifying these large quantities of text into a manageable list or set of concepts. Experience with this classification process has shown that it, itself, is a problem for facilitators (those meeting leaders responsible for the meeting process).

The EMS provides only clerical classification support for browsing the text and creating a list of topics for group members. However, it does not provide cognitive support for managing the large volume of text that may be created as output from an electronic brainstorming session. The synthesis of electronic brainstorming comments is a classification problem. It is something that humans currently do well -- but not willingly. This type of problem suggests using an artificial intelligence approach to understanding how humans classify concepts in an electronic meeting setting and then developing a prototype system to test whether better cognitive support for groups can be provided.

Chapter Two provides a description of the electronic meeting system and describes in more detail the classification problem that exists in the group problem-solving

process. Chapter Three surveys the literature concerning classification. Classification is viewed from a cognitive perspective -- modeling knowledge, summarization, and theories of abstraction -- and from a systems engineering perspective -- natural language understanding, neural networks, and self-organizing map systems. Chapter Four raises the research question, presents the system development methodology that guides this research, includes a description of the methods that human experts use in classification of electronic brainstorming comments, and describes previous work concerning the Hopfield Neural Net Classifier. Chapter Five describes this research -- the application of the Kohonen Self-Organizing Map (SOM) to the classification problem. Chapter Six describes an experiment that evaluated the classification performed by the Kohonen (SOM) as compared to human experts and the Hopfield Neural Network. Finally, Chapter Seven discusses conclusions, contributions of this research, and direction for future research.

CHAPTER 2: THE PROBLEM -- CLASSIFICATION IN EMS

Electronic Meeting Systems (EMSs) are a subset of a general category of systems, called "Groupware," that support Computer Support Collaborative Work. Ellis et al. define Groupware as "computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment" (Ellis, Gibbs, & Rein, 1991). In addition to Electronic Meeting Systems, other types of groupware include electronic mail, group calendaring systems, group project management, and multi-user document editing systems.

What distinguishes EMS from other types of groupware is that the common goal (problem resolution) typically involves complex, organizational problems which benefit from maximum participation from appropriate organizational members. Examples of organizational problems that benefit from the use of EMS include performance of organizational analyses through vision or mission statement construction or organizational model building, strategic plan development, organizational decision making, and project status reporting. Research and experience with Electronic Meeting Systems have demonstrated that electronic support of group meetings can improve meeting productivity through greater participation by organizational membership and shorter times for information gathering for group problem resolution (Nunamaker, Dennis, Valacich, Vogel, & George, 1991) (Post, 1992). In particular, groups using EMS have been found to generate more unique alternatives for creative tasks and that

quality of decisions related to intellectual tasks are better for EMS-supported groups than for non-EMS-supported groups (George, Easton, Nunamaker, & Northcraft, 1990).

Experience with EMS has demonstrated a consistent pattern of use in what may be described as a "group problem solving" process. This process is goal-directed in that specific types of information are identified for support of problem resolution. The EMS is then used to gather the information that is appropriate for the goal. We label the stages of this information gathering process as "divergence," "convergence," and "consensus checking." Particular EMS tools are selected for each of these stages depending upon the characteristics of the particular goal or task before the group. Electronic brainstorming (EBS) is a particularly good divergent tool for collecting information for the most complex tasks in which maximum, unstructured, and anonymous participation is desired. The Idea Organization (IO) or Categorizer tools might be used to organize the information into a list of topics or concepts that address the complex problem (the convergent stage). A Rank Order Vote might then be used to test whether all meeting participants agree on the definitions of those topics or concepts or, if the topics are well understood, whether participants agree on the importance of the topics with respect to the particular goal.

Divergent tools, EBS in particular, create a problem in the convergent stage due to the large quantities of ideas that they generate. Meeting participants experience cognitive overload in attempting to transform the hundreds of ideas generated during the divergent stage into a list of topics, concepts, or alternatives that address the particular group problem that was discussed in the divergent stage.

This chapter describes the EMS system used in this research, GroupSystems V, and discusses how it is used to solve group problems. In particular, Electronic Brainstorming (EBS) is explained in order to understand the convergence problems that occur in classifying the ideas generated in EBS to a manageable list of topics or concepts.

2.1 DESCRIPTION OF EMS

An Electronic Meeting System consists of hardware, specialized software, and facilitation methods and techniques for solving group problems. An EMS is a general problem-solving tool and can be used for a variety of problem types.

2.1.1 Description of Hardware, Software, and Facilitation

An Electronic Meeting System (EMS) typically includes 8-30 networked personal computers or workstations, special software that allows people to enter comments and manipulate shared data, and support for electronic projection of the shared data as well

Figure 1 -- Electronic Meeting System Facility

as other audio/visual equipment that support meetings. Figure 1 is an example of a facility designed for electronic meetings. Personal computers are arranged in two semi-circles. Built into the unit at the front of the room are two personal computers that a meeting facilitator uses for controlling the software that is used on each participant workstation. Two projection screens are centered in the front for projection of the facilitator screens when it is appropriate as well as connections to video tape and television tuners in a control room (not visible in the figure). White boards are

located on each side of the projection screens. The facilitator can control the lights and all audio visual equipment from the front during the course of the electronic meeting.

Figure 2 -- EMS Software Support

Software for general-purpose group problem solving consists of a set of programs that can be mixed and matched to suit the characteristics of the problem as well as the characteristics of the group of problem-solving participants. Figure 2 contains a diagram of the types of EMS Activities that are needed for an electronic meeting as

well as a mapping of particular GroupSystems tools that support the problem-solving activities.

Four EMS Activities diagrammed in the center of the figure are the main types of tools needed for group problem-solving: Idea Generation, Idea Organization, Consensus Polling, and Issue Exploration. Session planning software controls the overall meeting and is, itself, a set of software tools for creating agenda, starting and stopping participant tools, and system utilities such as saving data and printing reports. Since much of the data collected through use of the electronic meeting system is recorded, how the data is recorded during the use of any tool and used throughout the session is also a function of the session planning software.

GroupSystems is an specific example of specialized software tools that was developed at the University of Arizona. We focus on Electronic Brainstorming, Idea Organizer, and Categorizer in this research.

Facilitation is the activity of matching the appropriate EMS hardware and software tools to the organizational problem to be solved. This requires collecting information from organizational members to understand characteristics of the problem, characteristics of the group, and the organizational context of the problem. Figure 3 is an EMS Research Model discussed in (Nunamaker, Chen, & Purdin, 1991). Using this

Figure 3 -- Electronic Meeting System Research Model

model, facilitation is defined as the management of the central process that produces a specific desired outcome.

Characteristics of the Group, Task, Context, and EMS contribute to the process of solving a problem in order to achieve the desired outcome. The central component, Process, produces the desired outcome. The quality of the outcome is contingent upon the ability of the process manager to identify and address those characteristics of

Group, Task, Context, and EMS plus the characteristics of the desired outcome (Nunamaker, Dennis, Valacich, Vogel, & George, 1991).

Examples of Group characteristics include group size (large/small), group members' physical proximity (same place/different places), composition (range of backgrounds, interests, status in organization, etc.), and cohesiveness.

Task characteristics primarily deal with the level of complexity of the task. A simple task might consist of asking a group of participants to select a solution from a known list of solutions. A complex task might include having a group clarify and agree on the problem definition, generate as many potential solutions as possible, derive a list of solutions, and then prioritize the list of potential solutions.

Context characteristics include organizational culture, time pressure, reward structure and other factors which may influence how the group will work together.

2.1.2 Examples of Types of Problems Addressed by EMS

Examples of organizational problems include a lack of coordination among departments and/or organizational members, difficulty in making decisions for complex issues, or a need to better communicate across organizational departments or across

organizational hierarchies. Facilitation with EMS may support the production of a variety of products that help to overcome these organizational problems.

Strategic Plan. For example, a strategic plan may be a desired outcome that would address organizational coordination problems. A facilitator would meet with organizational leaders to identify elements that need to be in the strategy plan (task characteristics) as well as identify characteristics related to the appropriate people that should form the strategic planning group and background organizational information (context characteristics).

Organizational Modeling. EMS could also be used to analyze the organization by having organizational members create a model of the business processes. The model would only be part of the desired outcome. The analysis might also include a list of improvement ideas, redundant processes, and costs associated with the processes, for example.

Decision Making. Any decision of a complex nature that would benefit by participation by a large sample of the organizational population is a potential problem that EMS can support. Again, the facilitation of the problem solution would involve identifying the information that is needed to make the decision and understand as

much as possible about the group and context characteristics to select appropriate EMS tools to produce the desired outcome.

Project Status Reporting. Organizational members working on an organizational project may have a variety of views as to the actual status of large projects. An EMS can support rapid alignment of project team members, communicate status to the rest of the organization, and provide a forum for project planning.

Policy Formation. Organizational policies and regulations might be better formulated and supported if a greater number of organizational members took part in that formulation. An EMS can support a larger group of people more efficiently.

Many other types of problems can be supported by EMS. In any case, experience using an EMS to support problem resolution has shown that the desired outcome of an EMS-supported session is that specific information required to solve the organizational problem.

2.1.3 Typical Problem Solving Sequence

Experience with EMS in group problem-solving has shown that most use of the system is in a goal-directed approach; i.e., specific deliverables are identified that will support the problem resolution and those deliverables become goals for production by

the group with support of the EMS. Use of EMS in producing these deliverables has demonstrated a recurring three-stage pattern: 1.) Groups typically use parallel and anonymous data entry to brainstorm all of the issues that are related to a given deliverable ("get all of the issues out onto the table"). 2.) Groups review all of the issues to identify those which are most salient to the deliverable at hand. 3.) Depending upon the importance or sensitivity of the issues with respect to the group, the group may be anonymously polled to determine if there is group consensus as to the meaning of each of the issues and/or consensus with respect to relative importance among the issues. We refer to these stages as divergence, convergence, and consensus polling/voting, respectively.

2.1.3.1 Goal-Directed Approach. The desired outcome from the group's perspective is defined to be the ultimate deliverable of the group process (e.g., a strategic plan or a prioritized list of suggestions to present to upper management). From the research perspective, examples of outcome characteristics include levels of efficiency, effectiveness, and participant satisfaction with the process. From a practical, deliverable perspective characteristics of the outcome include the structure of the ultimate product (a document, ordered list, outline of concepts, collections of comments, etc.).

Figure 4 presents a model of this outcome decomposition. In the example of a strategic plan, Subgoal 1 might be a corporate vision statement. Subgoal 2 might be a mission statement. A strategic plan might also contain a list of corporate goals and/or

Figure 4 -- Decomposition of a Complex Group Problem

objectives. A prioritized list of goals might be Subgoal 3. A subgoal may also be something such as an analysis of corporate strengths, weaknesses, opportunities, and threats (SWOT Analysis). This subgoal could further be divided into four sub-subgoals -- each one deriving a respective list of strengths, weaknesses, opportunities,

and threats. Decomposition of the outcome typically stops when the objective is to derive a paragraph of text or list of items.

2.1.3.2 General Three-Stage Approach. Experience has shown a recurring pattern in the process of creating these paragraphs of text or lists of items with an EMS. Ideas are solicited to address the subgoal or sub-subgoal (divergence). Once ideas have been solicited, some order, synthesis, or classification of those ideas must be done (convergence). Finally, the group of people should be anonymously polled to determine some sort of consensus with the outcome of the convergence and its relationship to the desired subgoal.

Stage One: Divergence. For example, assume a subgoal of the strategic plan is a vision statement. How does one facilitate creation of a vision statement with twenty or more members of one organization? Depending upon the characteristics of the group, at least three methods are possible: 1) have each person write a vision statement -- thus creating 20+ "straw men" statements; 2) modify one "straw man" vision statement; and, 3) brainstorm on what elements need to be in the vision statement, get group consensus as to what those elements need to be, and use those elements to build the vision statement. Divergence in a group setting means that members of the group provide comments or ideas concerning the topic at hand. Divergence is also referred to as "*Idea Generation*." In an EMS facility, twenty or

more people will provide their comments in parallel. In method one above it is easy to see that the group may generate a large quantity of text if each person supplies a statement. In method two, it is usually appropriate to allow group members to provide anonymous comments concerning the sample vision statement. Here, again, group members might supply a generous quantity of comments. Method three builds the vision in a bottom-up manner.

Group members may anonymously suggest ideas for elements such as "customer satisfaction," "profitability," or "long-term growth". Again, twenty or more people might suggest a large number of elements in a short period of time. In each of these methods, divergence provides every member of the group opportunities to contribute to the sub-goal and, ultimately, to the output of the meeting. Anonymity of comments should remove any hesitancy of group members to contribute. Facilitators will choose the method of divergence that they feel will be most appropriate to the characteristics of the group, task, and context of the meeting. In each method, however, a large volume of text is created that needs to be synthesized in some way.

Stage Two: Convergence. Convergence refers to the synthesis of the text that is generated from the divergent stage of the problem-solving process. It is also referred to as "*Idea Organization*." The objective of the convergence stage is to produce a list or outline of topics that address the current sub-goal of the electronic meeting. The

intent is to categorize the comments or ideas found in the text into a topical list or list of concepts. In the vision example above, the objective might be to derive a list of elements that must be evident in a vision statement. In order to develop this list of elements, group members need to browse the comments produced during the divergent stage of any of the three methods described.

EMS tools provide support for browsing the text in a serial manner. However, synthesizing the complete set of textual comments into a list of categorical topics or outline of concepts that are contained in the text is performed in a bottom-up manner. Coordinating this effort among large groups is problematic. The EMS tools provide software allowing each group member browsing capabilities plus the ability to post topics to a public screen. However, groups have great difficulty in 1) determining the appropriate levels of abstraction of terms to be used in the synthesized topics; 2) maintaining consistency in the levels of abstraction; 3) not duplicating entries already on a list; 4) supporting topic suggestions with links to the ideas or comments that suggested that topic; 5) making sure all of the underlying comments have been reviewed for potential synthesis and addition to the list; and 6) taking the time necessary to go through all of the comments. The typical divergent stage takes about 40 minutes of time. The typical convergent stage takes 40 minutes to an hour.

Stage Three: Polling. The polling/consensus checking stage of the problem-solving process involves sampling members of the group to determine how much agreement exists concerning the topics that were created on the list created during the convergent stage. Voting tools allow for ranking, weighing, or otherwise evaluating list items. Output of the voting tools allows for analysis of the spread of the votes in the group -- again, anonymously representing the votes. Even if some method of polling or consensus checking is not required for the items generated from the convergent stage, meeting facilitators must check that the output list conforms to the requirements of the subgoal.

2.1.3.3 Tool Selection. Facilitation of an EMS requires matching the appropriate tool with the characteristics of the task, the group participating in the meeting, and context of the problem-solving session. Activities associated with the divergent stage map into the Idea Generation and Issue Exploration EMS activities of Figure 2. Convergent activities occur in EMS Idea Organization activities. Polling or consensus checking obviously correspond to the Consensus Polling activity of the EMS Model depicted in Figure 2. In the case of GroupSystems tools, the facilitator must know the EMS tools well enough to choose that tool which is most appropriate to characteristics of the task, group, and context.

Electronic Brainstorming (EBS) is an Idea Generation tool that supports the divergent stage for tasks which are complex and not well-defined. For example, a strategic plan might include identifying issues that will affect the organization's ability to function in the next five to ten years. It focuses on one complex issue in which participants have an interest. The tool also limits participant entering of text to five lines per comment or idea. The five-line limitation attempts to cause participants to think "broadly" rather than dwell on one specific topic or issue.

Other Idea Generation tools allow more text per comment and are appropriate for less complex issues and where a broader focus of attention is desired. Topic Commenter and Group Outliner allow for idea generation over multiple topics. Participants can generate ideas in those topic areas that are of interest to them.

Because of the complex nature of the comments generated through EBS and the focus on one topic, the output of this tool is most appropriate for this study.

2.1.4 Strengths of EMS

The benefits of group problem solving using an EMS come from three major EMS attributes: anonymous participant input, parallel data entry, and electronic recording of all user input.

Allowing anonymous input of information by participants may result in a freer exchange of ideas. If participants know that comments are anonymous they might be more willing to "get the difficult issues out onto the table." From the comment-reader's perspective, not knowing the source of a comment or idea removes the filter that people tend to apply in linking the source of a comment to a personality. A comment or idea that is separated from the identity of its source is more likely to be objectively evaluated with respect to the problem at hand.

Parallel data entry means that each participant is free to enter ideas or comments at the same time. Participants can enter information while ideas and thoughts are fresh in their minds.

Electronic recording of all of the participant input means that participants are free to enter comments and ideas while they are thinking of them yet still have time to read the input of others. This also means that a set of comments or ideas generated by participants in one EMS tool can be used as input to another EMS tool.

Anonymous participant input and parallel data entry are strengths that are most evident in the divergent stage of group problem solving. In the case of GroupSystems tools (Figure 2), the divergent tools of Electronic Brainstorming, Topic Commenter,

Group Outliner, and even Categorizer can generate hundreds of comments in a relatively short period of time (under one hour) with a group of twenty-five people.

Organizing the comments or ideas into some structure is supported in the GroupSystems EMS by Idea Organizer, Categorizer, and Policy Formation tools. Experience with EMS has shown that the strengths of anonymity, parallel data entry, and electronic recording are less effective in supporting the convergent stage of group problem solving.

2.2 CONVERGENCE PROBLEMS IN A GROUP SETTING

The objective of the Idea Organization activity in an EMS is to classify the issues or comments into a structure that is appropriate for the desired outcome, goal, or subgoal. Difficulties in performing this idea organizing activity are due to various aspects of the output of the divergent activity: large numbers of comments and different connotations surrounding some of the vocabulary contained within the comments are two types of difficulties. Other difficulties are related to the increased cognitive demand of the Idea Organization process itself as well as software interface problems with the idea organization tools.

An example of this convergence problem will be given below along with some facilitation approaches to help address the convergence difficulties. The facilitation

techniques are insufficient, however, and the need for better automated support will be described.

2.2.1 Difficulties with EMS Convergence

Problems associated with the convergent stage of group problem solving are associated with information overload, the vocabulary differences among the terms used in the comments, increased cognitive demand, and computer interface problems associated with managing large amounts of information with limited screen space.

2.2.1.1 Information Overload

Successful use of the Electronic Brainstorming (EBS) tool has been found to result in an unexpected information overload problem. In a typical meeting of 10-20 participants, several hundred EBS comments can be generated in less than an hour. The characteristics of parallel and anonymous data entry that the EMS supports coupled with the participants' close relationship with the problem create a reaction that generates hundreds of lines of text in a short period of time. This large quantity of text makes it extremely difficult for participants to browse comments and consolidate ideas. Meeting participants are often impressed with the number of ideas they have generated, but become overwhelmed with the task of organizing the ideas into categories.

2.2.1.2 Vocabulary Problem

Much of the complexity of interpersonal communication deals with semantics. Connotations surrounding terms differ from the perspectives of different individuals and, of course, even differ within the same individual in different specific problem settings. For example, it was our experience during a three-day meeting that two classes of people were in attendance: functional experts and technical experts in hazardous materials and waste management from the Armed Forces. It was assumed that functional experts were those people who were knowledgeable in the management of hazardous materials and wastes and that technical experts were the information systems people that created the information systems to support the hazardous substances management. It was midway into the second morning before the facilitator of the meeting realized that some of the functional experts in attendance who were chemical engineers were identifying themselves as "technical experts" and equating the software people as "functional experts" in supporting the chemical engineers in performance of their management functions.

Thus, one aspect of the vocabulary problem is that meeting participants may have different definitions of the same terms. Another aspect is the converse: multiple terms may have the same meaning with respect to meeting participants. For example, "Slippery Elm Army Post" might be also be referred to as "Slippery Elm," "SE Army Post," or "SEAP." In prior research (Furnas, Landauer, Gomez, & Dumais, 1983)

(Furnas, Landauer, Gomez, & Dumais, 1987), Furnas et al. found that in spontaneous word choice for objects in five domains, two people favored the same term with less than 20% probability. The probability of 10-20 people choosing the same terms in generating ideas within a specific domain has not been studied but would seem to be higher. In any event, meanings attached to specific terms in a comment must be considered in the complete context of the whole comment.

2.2.1.3 Increased Cognitive Demand

EBS comments need to be consolidated and organized by meeting participants within a short period of time. Usually each participant has to browse and understand the ideas generated in the EBS comments, judge the merits of these ideas, merge similar ideas, eliminate redundant or irrelevant ideas, consult other members' opinions, and so on in less than an hour. Because the EBS process encourages creative, diverse, and uncensored ideas, many EBS comments are raw or unpolished and it may require special effort from meeting participants to synthesize them. Because the group topic organization process is sometimes frustrating and sub-optimal, the satisfaction level and productivity of the meeting participants may go down significantly (discussed below), and/or some unique EBS ideas may be lost during the cognitively demanding process.

2.2.1.4 EMS Interface Problems

A 25-line computer screen is very constraining for most people to use in browsing hundreds of lines of text while trying to synthesize ideas into a list of issues. Several windows are needed: a window for browsing the comments, a window for formulating a participant's local list of issues, a window for viewing the public list (i.e., those local issues that have been submitted to the public list via participants' local lists), and a window for attaching comments to the list items either through attaching comments from the EBS file directly to list items or by entering new comments. Management of four windows is difficult for even computer experts let alone the average person participating in a meeting.

The user interface that participants use to attach comments to topic headings is particularly cumbersome. The intent is that participants might support the suggested topic or concept by copying the original comment that suggested the synthesized topic into an attachment to the topic. Participants can also type in their own comments to support the topic or further explain what it means. However, comments attached to topics on most lists are sketchy and incomplete. Details of this user interface are given in the description of the example below.

Finally, the current EMS Categorizer tool starts each member of the group in a different location in the text file. The assumption is that all of the comments should

be reviewed by someone in the group during the process such that there will not be a preponderance of topic list items addressing the comments at the beginning of the file and a dearth of topic items addressing those comments near the end of the file.

However, participants are generally confused that they do not all start in the same place in the file and require some orientation to the comment numbering scheme in order to feel comfortable that each person does, indeed, have the complete set of EBS comments and, thus, has equal opportunity to review each and every comment.

2.2.2 An EBS/IO Example

This example of the use of the Electronic Brainstorming (EBS) and Idea Organizer (IO) came from an actual session in which sixteen academic and industry people met to discuss future developments in Collaborative Technology. The desired outcome of this particular session was a rank-ordered list of Collaborative Technology problems to be solved in the next five years.

An EBS session was started by the facilitator from the Meeting Manager software. The EBS question was entered: "What are the most important information technology problems with respect to Collaborative Systems to be solved over the next five years?". After the facilitator confirmed that all of the participants understood what the question meant, he started the participant stations. Each participant had an EBS entry screen similar to Figure 5.

Electronic Brainstorming

Prior Comments: - IT Problems for CSCW ————— Top Line 1 of 4

=====

What are the most important information technology problems with respect to
Collaborative Systems to be solved over the next five years?

=====

————— Your Comments <limit of 5 lines>: —————

2.1

F1:Help F2:Send Alt+F2:Exit

Figure 5 -- GroupSystems Electronic Brainstorming Participant Screen

The participant screen depicted in Figure 5 shows two windows -- a "public area" at the top and a black "private area," at the bottom. The public area contains the EBS question and will contain anonymous comments made by other participants as the discussion continues. The private area is where the participant is instructed to add his or her comment -- in this case, one Collaborative Technology problem per comment. The private window is limited to five lines of text. Note also from Figure 5 that the system has automatically provided a comment number for this participant's comment in the private window (2.1). This means that the comment to be entered will be the

first comment of file number two. When the EBS tool is started by the facilitator, each participant receives a unique file plus the system creates at least one additional file. When the participant types in a comment and "Submits" it by pressing the F9 key, the comment is attached to the bottom of the file and is sent to the pool of available comments. The participant who submitted the comment will receive another file at random from the comment pool and has the option of commenting on the comments contained within this new file or adding something completely different.

Meeting participants typically diverge using the EBS tool for forty minutes to one hour. They typically generate hundreds of comments. Figure 6 shows a sample of the output of an EBS session. The output is a concatenation of each of the EBS files (in this case there were 30 files). File number 1 has four comments. File number 30 contains seven.

This example represents the exact output of the Electronic Brainstorming session except that the actual name of the company referenced in the comments was replaced by XYZ.

The example shows that the four comments associated with File 1 (1.1 - 1.4) all discuss the topic of standard user interfaces. The comments numbered 30.5 and 30.7 in Figure 6 seem to be talking about different concepts than the integration issues that

```

Electronic Brainstorming Report
Session: technology Date: 11/18/1993 2:08pm
-----

=====
What are the most important information technology problems with respect to
Collaborative Systems to be solved over the next five years?
=====

1.1 development of "standard" user interfaces for remote access to info
and communication

1.2 Doesn't XYZ Software give us such a standard?

1.3 XYZ is one implementation, but it certainly is not yet a standard.
!!!! Maybe we need to have a standard that is NOT owned by a company which
has a history of initiating look and feel lawsuits when someone else copies
their "standard" !!!!!

1.4 XYZ is not a good standard even if they are the industry leader
at this time. XYZ is a terrible architecture.

2.1 Voice recognition is one of the promising technologies.

...

30.3 Understanding how we actually use the tools in a distributed mode.
We need to understand process.

30.4 Total integration between all applications.

30.5 how to measure the validity and reliability of groupware tools

30.6 Yes, total integration. Right now artifacts of the tools limit us
significantly in ways we need not be limited.

30.7 No matter what we do we will be limited in ways we need not be
limited. Thank God for unanswered prayers.

```

Figure 6 -- Sample EBS Output

are discussed in the other comments.

Figure 6 also demonstrates that the data can be "noisy" with typographical errors, abbreviations, and jargon.

The goal of the convergent process is to synthesize the file of comments represented by Figure 6 into a list of Collaborative problems or problem types. The

GroupSystems Idea Organizer tool is one method that facilitators can use to create this list.

Figure 7 -- GroupSystems Idea Organizer Beginning Screen

Figure 7 presents the participant view of the beginning of the Idea Organizer tool. There are two windows. One is labeled "Public - Collaborative Tech. Problems" (top) and the other is labeled "Reference File" (bottom). The Public window will contain a list of Collaborative Technology Problems or a list of types of Collaborative Technology problems that the group will identify by browsing the text that is in the

Reference File window. The reference file is, in this case, the output from the Electronic Brainstorming session. The first comment is highlighted (1.1 development of "standard" user interfaces ...). The two lines comprising the first comment are the first two of 576 lines as evidenced by the indicator in the upper right portion of the window. Users browse through the reference file by using the PgUp and PgDn keys on the keyboard. If a participant wants to add a problem or type of problem to the list in the top window he or she presses the down arrow or the INS key.

Figure 8 -- GroupSystems Idea Organizer Add Entry Screen

Figure 8 is the screen that the participant uses to add an item to the public list. The Reference File window has been replaced by two windows: "Item:" and "Your

Comments:". In this example, the participant has generalized the "development of 'standard' user interface" contained in the first comment to "standards." The Item window has enough room for one short line of text. The purpose is to restrict the participants to bullet-type of items or issues. The bottom window is to add explanatory text to the item or issue should the participant want to expand on an explanation of what the item means. Participants also have the option of attaching the original comment that is highlighted in the Reference File should they wish to. However, the currently-highlighted Reference File comment is hidden from view.

The Idea Organizer screens in Figure 7 and Figure 8 demonstrate the tool with the "Private Screens" option off. The tool allows for a duplicate set of screens where participants can build lists without any other participants seeing their work. If a list of items with or without attached comments are built in a private window they must then be marked by placing the highlight bar on the list item, pressing the space bar, and then pressing the F9 key to submit the item to the public screen. Participants can change between the Public screen view and the private view by pressing the F6 key.

Much of the dissatisfaction associated with initiating the convergent process is based upon the realization that, in order to do the task, the large body of text must be reviewed one sentence or one comment at a time. Several facilitation techniques evolved to support this "bottom up" approach to categorization.

2.2.3 Techniques for Facilitating Group Idea Organization

The frustration experienced by meeting participants in association with the cognitive complexities of organizing EBS ideas is significant. Experienced facilitators have devised a number of group techniques or have used EMS tools in a variety of ways to attempt to reduce this frustration.

2.2.3.1 Avoid EBS Complexity Completely

One technique is to avoid the EBS tool entirely. If the problems of information overload, vocabulary, and high cognitive demands are due to the highly unstructured characteristics of EBS then some facilitators opt for more structured tools such as Topic Commenter, Group Outliner, or even Idea Organizer, itself. The stress felt by facilitators in trying to coax participants into identifying issues at the beginning of the convergence stage is particularly high when the facilitators are unfamiliar with the group problem domain. It is possible to work with group leadership to collect meeting participant comments and produce the desired outcome without using EBS. However, it is an open research question as to whether this inhibits generation of innovative ideas that may result from an unstructured EBS session and encourages potential "group think" that may result from foisting existing leadership structured discussion onto meeting participants.

For actual EBS output, several facilitation techniques can be used to address convergence problems:

2.2.3.2 Divide and Conquer Method

The set of comments can be partitioned into a number of sections equal to the number of participants. Participants can derive lists of topics related to the comments within their assigned sections and these lists can then be merged. If each of twenty-five participants submits five suggestions a list of 125 topics result. The facilitator must then lead the group in browsing this list using a public screen. Since a target "manageable" list of topics contains 10 to 20 topics, a great amount of time is spent in leading the group in eliminating redundancies among the 125 topics and leveling the topics to a level of abstraction which will produce a reasonable number of topics that are appropriate for the EBS question. Leveling the topics refers to eliminating topics that are too general and merging together topics that are too specific with respect to the current goal. However, a list of 125 items can, itself, be an overload of information for group consideration.

2.2.3.3 Limit the Input

One technique used to control the number of suggestions is to attempt to focus the group first on an appropriate level of abstraction. A variation of the Nominal Group Technique (Fox, 1989; Frankel, 1987; Dennis, 1991) might be used whereby

participants can be asked to propose one topic each. One suggested topic can be collected from each group member anonymously and posted to a public screen. The group can then browse this list on the public screen and evaluate the items with respect to relevance to the sub-goal and level of abstraction. The facilitator guides group members in looking for duplicate entries and topics that are too general or too specific with respect to the current goal. If the group continues to have trouble settling in on a level of abstraction, another round of 1-item suggestions can be collected and evaluated as a group.

Duplicate topic items or synonymous topics can be merged together under the topic heading that seems most appropriate to the group. Items that are too general or inappropriate are typically deleted. For example, in the vision example above, a participant may suggest "maintain business in a legal manner" as one of the elements that need to be contained. This may be a topic that is simply understood and too general to be used in a given organization's vision statement.

It is usually desirable to support a topic with comments that enrich its semantic content. Participants can either attach the original brainstorming comment that suggested the topic or they can type in text that further explains the topic. Experience has shown that group members typically do not attach much support to the topic suggestions. Having just discussed the issues in EBS, most everyone understands the

topic. However, as part of the validation of the outcome of the group effort, it would be better to be able to trace through the logic of the electronic meeting with topics that are directly linked to the comments that were used to generate them.

2.2.3.4 Keyword Option in EBS with Group Input

A facilitation technique which is used to automatically create this link takes advantage of the keyword option of electronic brainstorming. After participants are finished generating ideas the facilitator leaves the group within the EBS tool and asks people to verbally suggest topics from the individual files that are displayed on their screens. The facilitator types in the topics and adds keywords and synonyms of keywords to the topic. The system then searches the file of EBS comments and attaches all of the comments containing the keywords to that topic. This technique cannot be done anonymously but can quickly get the group started on a list of topics that can then be loaded into the Idea Organizer tool for anonymous comment and addition.

2.2.3.5 Keyword in EBS with Facilitator/Leader Input

Finally, the facilitator might begin the convergent task while the group is still diverging in electronic brainstorming. The facilitator can browse the comments as each comment is submitted by the participants and begin entering terms that typify the content of any particular comment or collection of comments. When the group is

finished brainstorming, the facilitator can then use this list of keywords as a starting list for idea organization. Results of this technique are highly dependent upon the facilitator's knowledge of the electronic brainstorming domain. Further, because comments from 20 to 25 people can come in rapidly at times it is difficult for the facilitator to completely read each comment and consistently apply keywords.

Each of these facilitation techniques approaches the problem in a bottom-up manner. Problems associated with these approaches include:

- Inconsistent analysis of the EBS comments. It is typical that not everyone reviews every comment.
- Groups do not tend to provide links from the topics to those comments that support the topics.
- It is slow and difficult. Groups require as much time or more to organize the ideas as they took to create them (1 hour is typical).

2.2.4 Desired Classification: The Topic List

With any technique, the objective of the idea organization process is to arrive at a list of topics representing the contents of the set of EBS comments.

Figure 9 provides a sample of what a group might create as a result of organizing the ideas generated in the EBS output of Figure 6. The list contains thirteen topics

Sample Topic List	
1.	video projection
2.	network bandwidth / virtual reality
3.	multimedia / hypertext
4.	group memory / project memory / repository
5.	voice recognition
6.	organizational culture / style / language differences
7.	standards
8.	distributed work / distance issues / facilitation / project management
9.	research methodologies
10.	costs
11.	team / integration / social norms
12.	performance / reward structures
13.	education / training / learning

Figure 9 -- Sample Solution to Group Classification Problem

that are related to collaborative technology problems to be solved in the next five years. Note that this list is not a list of pure technology problems. The list provides a high-level abstraction of the topics that were discussed in response to the EBS question.

Ideally, members of the group would have linked comments to each of the topics or they would have attached text that support further descriptions and/or definitions for each of the topics. If keywording is used, the system links those comments containing the identified keywords to those topic list items.

Each of the thirteen topics consists of one or more *terms* or a *term phrase*. Topic One, "Video projection", is a term phrase that was identified to be a collaborative technology problem area. "Network bandwidth" and "virtual reality" are two term phrases that, though separated by a "/", need to be considered together as one topic (Topic Two). One might interpret this problem area as "network bandwidth to support such things as virtual reality" or as "virtual reality problems such as bandwidth." Ambiguities such as this may be resolved in the comments that are attached to this topic.

Also, it is not necessary that the particular term or term phrase appear in the comment itself. Participants may paraphrase the comment to align it better with the EBS question or they may choose a more general term or term phrase to better capture the meaning of a comment as it applies to the EBS question.

Thus, each of the topics on the list is a cluster of terms and term phrases which form a conceptual cluster that addresses on aspect of the EBS question. We will use "terms" and "term phrases" synonymously in this paper as components of "topics." Since each topic is a conceptual cluster of terms, we will also use "topic" and "concept" synonymously.

2.2.5 Need for Automated Support

Figure 10 -- Top-Down and Bottom-Up Process

Figure 10 contains a hierarchical view of the divergent and convergent process. The EBS question sets the scope of the discussion. Divergence creates a set of EBS questions where each 5-line comment is a unit. The goal of convergence is to transform the comments into a list of topics or concepts which address the original EBS question. These topics consist of term clusters where the terms may or may not be contained within the comments.

Divergent activities for group problem solving are fairly well supported and are well aligned with the parallel data entry and optionally anonymous input strengths of EMS. The convergent activity, however, is not so well aligned nor well supported. The need for automated support arises from the dissatisfaction experienced by meeting participants, the inconsistent linking of comments to topics, the time required for convergence, and consistency of comment analysis.

2.2.5.1 Low Group Satisfaction. People enjoy expressing their opinions -- whether or not they participate in the electronic discussion associated with EMS. However, the overwhelming volume of brainstorming comments that are produced using electronic meeting software create a burden for meeting participants and facilitators during idea organization. This process becomes a bottleneck that counteracts the productivity gains and reduces the satisfaction of meeting participants.

Figure 11 presents a view of the satisfaction level through the group problem-solving process (Chen, Hsu, Orwig, Hoopes, & Nunamaker, 1994). The horizontal dimension reflects the typical EMS problem solving sequence. The vertical axis represents the satisfaction levels of the group.

Experienced facilitators locate the point in the process where satisfaction drops as that point when participants are confronted with the Idea Organizer screen depicted in

Figure 11 -- Satisfaction level as a function of tool usage

Figure 7. The exhilaration experienced by meeting participants quickly diminishes when confronted with the large number of comments that were generated. The discussion within the comments occur at a large variety of levels of abstraction -- from very specific issues, ideas, problems, etc. to very general -- and it is difficult to know which level is appropriate for synthesis. In other words, hundreds of issues or ideas may have surfaced during the EBS discussion concerning the one topic suggested by the EBS question. A manageable list of topics must exist somewhere between the hundreds of issues and the one EBS question (Figure 10). A list of potential topics at

this point in the process would be better than the empty list that participants typically face.

Facilitators use the techniques described above to try to minimize and/or guide participants through this frustration. Eventually, the point will arise when the group has offered topic suggestions, merged duplicate (synonymous) topics, deleted topics that are too general, and merged topics that are too specific in with more general topics. When a core of appropriate topics have been identified, satisfaction levels increase.

One benefit from the frustration that is experienced by meeting participants is sense of ownership that participants acquire once the process reaches the point of Further Group Problem Solving. This would suggest that if some agent, human or otherwise, proposed a list of topics that some discussion of those topics would be necessary or validation of the suggested topics would be required. Validation may come in the form of links to the comments that support the topics.

2.2.5.2 Linking Comments to Topics. The linkage of comments to the topics on the topic list is a form of validation, explanation or audit support. By definition, the problems that the group are trying to solve are complex. Support for definitions of the terms and the topics that these terms create help clarify the meaning of the topics.

The meaning of the topics is most often captured in participants' minds as they verbally discuss the topic list and determine which items are redundant, too specific, and/or too general. The connotations associated with the topics typically does not get recorded. People do not take the time to attach supporting comments to items on the topic list. Topics which remain unclear are discovered after consensus polling by looking at the spread of votes for all of the topics. "Fuzzy" topics are reflected by vote spreads across the voting dimension. Facilitators and participants must decide on the relative importance of further defining these topics with respect to the problem at hand.

Better linking of the topics with the EBS comments that produced them may allow participants to easily edit the comments and provide more specific definitions of the topics.

2.2.5.3 Shortening Convergence Time. Experience has shown that time of convergence is equal to or longer than the time of divergence. This convergence phase itself consists of two stages: topic identification and list resolution. Automating the topic identification phase may shorten the period of time required to browse the entire set of comments.

2.2.5.4 Consistency in Topic Identification. Identification of topics is typically done by participants browsing the comments in an ad hoc manner. Topic suggestions may be heavily favored to those comments which are near the beginning of the EBS comment file. Suggested comments also reflect points of views of the participants. Participants with stronger viewpoints or faster fingers may heavily populate the topic list with topics supporting their viewpoints. Consensus polling will typically catch this. Automating the identification of topics might provide a complete and consistent analysis of the comments that might support the convergence process.

2.2.6 Desired Solution: Automatic Classification of EBS Comments

Understanding more about the human classification process that occurs during this convergent stage helps in developing software that might produce an objective list of potential items that meeting participants could use to start the convergent stage. The objective of this research is to achieve this greater understanding, develop a more proactive groupware software component, and evaluate the use of this component.

A literature review was performed to determine the current level of understanding of the classification process and to determine to what extent existing information systems can perform classification of noisy textual data similar to that found in Electronic Brainstorming output.

CHAPTER 3: CLASSIFICATION LITERATURE REVIEW

Classification of textual data is very near to the root of how meaning is attached to words. To apply automated classification techniques to Electronic Brainstorming data requires both a search of the cognitive science literature and the system engineering literature.

3.1 COGNITIVE VIEW

A literature review of the cognitive literature results in three cognitive perspectives on classification: modeling knowledge, summarization, and abstraction.

3.1.1 Modeling Knowledge

In (Belkin, 1990), Belkin discusses the cognitive view as it pertains to information science. The purpose of the essay was to honor B.C. Brookes, one of the earliest proponents of the cognitive view of information science. In this essay, Belkin credits de Mey in directing a cognitive viewpoint "that any *processing of information* ... is *mediated* by a system of categories or concepts which, for the information-processing device, are a *model* of his *world*." To quote Belkin himself, "the essence of the cognitive viewpoint is that it explicitly considers that the states of knowledge, beliefs and so on of human beings (or information-processing devices) mediate (or interact with) that which they receive/perceive or produce."

(Lakoff, 1987) states that cognitive science is in a state of transition from the classical view to a modern view. In the classical view, categories are defined in terms of common properties of their members. In this view, human thought is basically symbolic manipulation. However, the modern view defines categories in a broader sense. Categories are much more than clusters of items with common properties. Categories serve as collections of conceptual systems and human thought goes far beyond the manipulation of symbols. Lakoff further states:

"Categorization is not a matter to be taken lightly. There is nothing more basic than categorization to our thought, perception, action, and speech. Every time we see something as a *kind* of thing, for example, a tree, we are categorizing."

Building upon the work of Eleanor Rosch, Lakoff describes categories as having *prototypes* and that there is a hierarchy of categories with the existence of a base-level category. The category of "bird" may be basic and, consequently, a common set of properties may be found in the human mental representations. Attributes such as feathers, wings, beaks, eggs, etc. are properties that we expect are basic to most human mental representations of a bird. Lakoff and Rosch's point are that experiences with birds attaches many more properties to each human's conceptual model of a "bird" than only the basic properties.

(Tanaka & Taylor, 1991) extended the research of Rosch and Lakoff to show that changes in the structure of classification hierarchies in object categorization occur with gains in expertise. Expertise deepens the hierarchy in that subordinate level categories become as differentiated as the basic level categories, subordinate level names are used as frequently as basic level names for identifying objects, and subordinate level categorizations are completed as quickly as basic level categorizations.

(Latta & Swigger, 1992) defines cognitive modeling as referring to any of a number of psychological theories developed to describe the ways in which individuals may employ internally constructed models of the external world to efficiently interpret and predict events in their environment. Other theoretical structures used to represent knowledge include schema, cognitive maps, frames, scripts, schemata, and semantic nets. She selected the repertory grid as a knowledge structure because she felt it offered the advantage of ease in creating subjective models. Her experiment showed that student subjects created similar repertory grid results for two concept areas. She felt that this validated the repertory grid as a knowledge structure. The repertory grid is interesting as a knowledge structure because it is created by differentiating triads of items instead of clustering items based upon similar properties. Latta referred to these properties as "constructs." The experiment consisted of two stages, subjects went through list of ten components three at a time differentiating only on the construct elicited from the subject as a difference for the first triad. In the second stage, an

expert supplied a list of 10 constructs that were created among all 10 triads by the expert. It would seem to be more interesting to have noted the differences in repertory grids that would have been created had each subject built all 10 constructs.

A formal mathematical model for conceptual knowledge systems is described in (Luksch & Wille, 1990). The authors adopt the notion from traditional philosophy that there are three basic notions of conceptual knowledge (objects, attributes and concepts) which are linked by four basic relations: an object has an attribute, an object belongs to a concept, an attribute abstracts from a concept, and a concept is a subconcept of another concept. An example of a conceptual knowledge system of zinks (a musical instrument) is used to demonstrate the model.

The literature in modeling knowledge supports the hierarchical representation of the group problem-solving process represented in Figure 10. Viewed from this perspective, group problem-solving is a process in which multiple people with all of the diversity of understanding that is contained within the group must converge upon one hierarchical model of the problem. This supports the experience of ease with which groups diverge or discuss the issues since individuals are free to present their mental models of the problem. Convergence is the attempt at arriving at one view at an appropriate level of abstraction.

3.1.2 Summarization

One facilitation technique in the convergent process is to ask participants to summarize each comment. This summary or label might then serve as a topic applicable to the problem at hand.

Richard Alterman (Alterman, 1991) states that "the capacity to summarize is a fundamental property of intelligence." He also says that the summarization results provide a test of a given model of understanding. He implies that there are various "models of understanding." For example, a thematic level of understanding would be indicated by someone summarizing a narrative in terms of themes. He does not present any other models of understanding but hints at a few by listing various types of summaries: abstracts, epitomes, overviews, abridgments, digests, and recapitulations. Alterman reviewed the literature on understanding and summarization and derived (summarized) four critical features of a summary: coherence (the summary must make sense), coverage (the summary must cover the bulk of the original understanding), importance (the summary must include the important features and exclude the unimportant ones), and workload (the summary is a simplification). From an artificial intelligence viewpoint, he defines a machine as modeling understanding when it can produce a representation that includes not only the initial elements of the text but also includes a representation of the connections among the elements. Machine understanding thus implies correct knowledge representation.

Analysis of textual data is a necessary step in doing qualitative research. Anselm Strauss and Juliet Corbin (Strauss & Corbin, 1990) recognize the fact that qualitative research has many connotations. To them, any technique of coding data that allows for statistically analyzing the codes is not a pure qualitative research process. A grounded theory is defined as one that is inductively derived from the study of the phenomenon it represents. Unlike the summarization process described by (Alterman, 1991), Strauss and Corbin differentiate between conceptualizations of the data that are thematic summaries versus conceptual interpretations of the data which create theory. Grounded theory looks for patterns in the textual data and apply conceptual tags to these patterns. It also involves relating these conceptual tags to each other. To Strauss and Corbin, description or summarization is merely removing some of the terms in the textual data. They define "concepts" as conceptual labels placed on discrete happenings, events, and other instances of phenomena. "Category" is defined to be a classification of concepts that is discovered when concepts are compared one against another and appear to pertain to similar phenomenon. Thus "categories" are at a higher level of abstraction. Categories have attributes and can have dimensions associated with the attributes. Categories can also be hierarchical such that categories may be clustered into parent categories (Strauss & Corbin, 1990).

Renata Tesch creates a spectrum of four qualitative research types which analyze qualitative data for a range of purposes. The most structured type analyzes textual

data for characteristics of the language. Potential interests of the researcher include understanding language as communication or as a cultural phenomenon. The second type is consistent with Strauss and Corbin in that it seeks the discovery of regularities in the qualitative data for identification and categorization of elements (Strauss's "concepts") and connections as well as higher-level patterns. The third type of analysis seeks to comprehend the meaning of the text or action through the discovery of themes or through interpretation of the data. The fourth type is the most holistic -- reflection. Reflection type analyzes the qualitative data for its literary content or meaning as a complete work. Tesch surveyed software products for use in analysis of qualitative data. Software products that claimed to be text analysis packages were merely text coding packages; i.e., they supported two text windows for files (one for the raw text and one to allow linkage of the researcher coding to the raw text).

Software products in (Fielding & Lee, 1991) also represent textual data bases and methods of using word processors as coding software without any reference to software that actually aids in directly analyzing the text other than keyword indexing programs.

A different approach to text summarization was applied at EDS Research (Shuldberg, Macpherson, Humphrey, & Corley, 1993). The algorithm filters text into a *TemplateFiller* that produces standard form bulletins from raw text input from

computer industry journals. The template must be set to match the type of information desired and is limited to consistently formatted input.

3.1.3 Abstraction

(Giunchiglia & Walsh, 1992) informally describes abstraction as "the process of mapping a representation of a problem, called the 'ground' representation onto a new representation, called the 'abstract' representation which (1) helps deal with the problem in the original search space by preserving certain desirable properties and (2) is simpler to handle as it is constructed from the ground representation by 'throwing away details'." The authors state that their main goals of this ambitious work are twofold: The first goal is to provide the foundations of a theory of abstraction which can be used for modeling and representing reasoning with abstraction as performed in common sense reasoning and by AI computer programs. The second goal is to use the theory as the basis for the development of a general environment for the use of abstraction in automated deduction (theorem proving and formalizing common sense reasoning). Authors provide an extensive mathematical proof of their theory and examples of case studies of use of application of their theorem-proving studies.

One assumption in layering of abstractions is that terms are more specific the lower they are in the levels of abstraction versus less specific at higher levels of abstraction.

3.2 SYSTEMS ENGINEERING VIEW

A literature search for software techniques for classification of textual data required review of the textual analysis literature as well as the literature on Cluster Analysis.

Within textual analysis, the automatic indexing and natural language processing literature was reviewed. Cluster Analysis divides into the tradition serial approaches used in statistics and the newer, parallel approaches using neural networks.

3.2.1 Textual Analysis

In order to analyze text using automated techniques one has to keep track of the terms of that text. The listing of the terms contained within the text and the locations within that text where those terms can be located is an index. Large bodies of text or textual collections suggest a need for using automated indexing techniques. Thus, the automatic indexing literature was reviewed to determine applicable techniques for text of the type found in electronic brainstorming comments.

Deriving concepts from the text seemed to imply some understanding of the text. Thus, the natural language understanding (NLU) literature was also reviewed to determine if techniques used in this area might be useful for deriving concepts from electronic brainstorming text.

3.2.1.1 Automatic Indexing

The classic explanation of automatic indexing is found in (Salton, 1989). Salton defines *indexing* as the process of constructing document surrogates by assigning identifiers to text items. *Manual indexing* is performed by human experts that are knowledgeable in a subject area. These human indexers frequently utilize indexing aids such as auxiliary schedules in the form of terminology lists and "scope notes" describing allowable vocabulary entries and giving instructions for the use of terms (Salton, 1989). The human indexers use their knowledge of the subject area to interpret the terminology of the document and classify the document according to the terminology dictated by the scope notes. Synonymous terminology of documents are interpreted by the human indexers and the controlled terms of the scope notes are used to index the documents.

Automatic indexing is defined as the machine performance of the process of constructing document surrogates by assigning content identifiers to text items (Salton, 1989). Information retrieval in the Information and Library Sciences has involved the indexing of documents for the purpose of both searching of specific documents and for browsing documents of similar topic areas.

Less control of vocabulary is possible using automatic indexing. This results in a greater variety of index descriptions. Synonymous indexes may result such that a

greater number of automatically-indexed terms may be needed to retrieve the same set of documents that would be retrieved with fewer terms in a manually-indexed set of documents.

Measures for determining the effectiveness of the indexing system (content analysis) include *indexing exhaustivity* and *term specificity*. Salton defines indexing exhaustivity as the degree to which all aspects of the subject matter of a text item are actually recognized in the indexing product (Salton, 1989). More exhaustive indexing implies a greater amount of the subject area will be reflected by corresponding term assignments (even minor aspects of the subject area will be recognized). Term specificity measures the degree of breadth or narrowness of the terms. Broader terms may index a greater number of documents but some of the documents may not be relevant or pertinent. Narrower terms index fewer documents that may all be relevant. However, a larger number of narrower terms may be needed to retrieve relevant documents. To measure indexing exhaustivity and term specificity two parameters are used: recall (number of relevant items retrieved divided by the total number of relevant documents in a collection) and precision (number of relevant items retrieved divided by the total number of items retrieved) (Salton, 1989). A trade-off exists between terms which are too specific and ambiguous terms which may index documents which are not necessarily conceptually consistent.

Salton describes three theories for indexing terms to aid in discriminating among terms to better address term specificity: term-frequency, term-discrimination, and probabilistic term weighting. Each of these three help to address single term measures for determining specificity/ambiguity. However, greater information content is found in the grouping of terms such as noun phrases that occur in the documents. The term phrase permits reasonably unambiguous interpretations but specificity must still be controlled through the use of thesauri or other hierarchical term arrangements (Salton, 1989).

Three blueprints for automatic indexing are described in (Salton, 1989). Two of these blueprints depend upon the existence of a thesaurus. Since electronic brainstorming text is generated in "real time" during a meeting and since most organizations do not have a documented "organizational thesaurus" we borrowed this five-point blueprint from Salton:

1. Identify the individual words occurring in the documents of a collection.
 2. Use a *stop list* of common function words (and, of, or, but, the, etc.) to delete from the texts the high-frequency function words that are insufficiently specific to represent content.
 3. Use an automatic *suffix-stripping* routine to reduce each remaining word to *word-stem* form; this reduces to a common form all words exhibiting the same stem (for example, analysis, analyzer, and analyzing are all reduced to stem ANALY).
-

4. For each remaining word stem T_j in document D_i , compute a weighting factor w_{ij} composed in part of the term frequency and in part of the inverse document-frequency factor for the term, for example, $w_{ij} = tf_{ij} \cdot \log(N/df_j)$.
5. Represent each document D_i by the set of word stems together with the corresponding weighting factors, that is, $D_i = (T_1, w_{i1}; T_2, w_{i2}; \dots; T_r, w_{ir})$.

(Fox, 1992) defines automatic indexing as the process of algorithmically examining information items to generate lists of index terms. "The lexical analysis phase produces candidate index terms that may be further processed, and eventually added to indexes (Fox, 1992)." The emphasis here is on how to parse the document, identify terms, remove stop words, and generating the lexicon. This type of analysis is useful for identifying terms and preparing text for content analysis.

Support for using term phrases came from experiments reported in (Faraj, 1994). A corpus of software engineering documents was used to evaluate automatic indexing methods using a combination of syntactic text analysis with statistical methods. A systematic performance improvement was shown in using composite terms generated by syntactic analysis in comparison to using simple terms.

(Ginsberg, 1993) describes the design of an automatic indexing system based upon two cooperating expert systems developed at AT&T Bell Lab. A statistical preprocessing module identifies the content-bearing words. An expert system named

Topic Expert determines the subject areas of the document from these words by matching them to vocabularies characteristic of given subject areas. Another expert system, the Subject Specialist uses a thesaurus and assigns final index terms according to the subject areas and the content-bearing words.

The literature indicates that content analysis of textual data using automatic indexing techniques requires the existence of a thesaurus or similarly prepared list of terms to guide the analysis. The analysis of electronic brainstorming text is a real-time effort that cannot benefit from such a thesaurus. However, the automatic indexing blueprint as described by Salton was used as a model to index the electronic brainstorming comments.

3.2.1.2 Natural Language Understanding

Electronic brainstorming comments are textual comments that participants enter in response to a brainstorming question. Deriving concepts from these comments seemed to require understanding the natural language contained within those comments. Therefore, a review of the natural language understanding literature was made.

In an artificial intelligence approach to natural language processing (NLP), (Rich, 1991) separates NLP into two tasks: generation and understanding of text. The generation of language involves the use of computers to create text in a conversant

mode with a human in such a way that the human could not determine that he or she was conversing with a computer (the well-known Turing test). Language generation also involves the translation of text from one language to another.

On the other hand, natural language understanding is involved with the internal representation and processing of text (spoken or written) in a manner that demonstrates cognition typifying a human. Identification of concepts contained within electronic brainstorming comments would seem to demonstrate some level of natural language understanding.

(Rich, 1991) outlines five steps in the understanding process: morphological analysis, syntactic analysis, semantic analysis, discourse integration, and pragmatic analysis.

Morphological analysis must separate the input text into terms and/or term phrases. Syntactic analysis must parse the output of morphological analysis and apply a sentence structure or other grammatical structure to the input. Semantic analysis must derive meaning from the input by mapping the terms into appropriate objects in a knowledge base and convert the syntactical structure into structure consistent with that represented in the knowledge base ("frames" or "scripts," for example). Linking these structures together to understand the linkages between the structures created in

semantic analysis (e.g., link pronouns of one sentence with their antecedents of previous sentences) is the objective of discourse integration. Finally, actual interpretation of the input and determining what action to take in response to the input is the objective of pragmatic analysis.

Work in syntax and grammars of English text has been occurring for decades. Noam Chomsky is best known for his work in syntax theory (Chomsky, 1965). Context-free grammars attempt to identify the building blocks of language with syntax providing the rules of construction. Construction is limited to phrases and sentences. (Woods, 1972) describes the augmented transition network (ATN parser) -- a system similar to a finite state machine of states and arcs where the states of the network represent positions through a text passage and arcs represent possible words, word categories, or procedures which cause a transition to the next state. An ATN parser still involves only syntactic analysis.

Semantic grammars (Burton, 1976) include semantic rules as well as syntactic functions in grammar rules. Rather than embed the semantics within the grammar rules, (Fillmore, 1968) creates semantic relations to structure the syntactic rules and called this a case grammar. Thirty years of work in syntax and semantics have demonstrated the complexity of processing the English language. There is still much

work to do to create semantic analysis output in a generalizable and useful way for discourse integration and pragmatic analysis.

Paul McKeivitt provides a good explanation of the state of NLP and a paper on "Approaches to natural language discourse processing" in (McKeivitt, Partridge, & Wilks, 1992). He describes the major problem in NLP as building theories and models of how individual utterances cling together into coherent discourse. We do not yet know how to capture what makes a given discourse rational. Early NLP was concerned mainly with syntax of natural language. In the early 70s Yorick Wilks and Roger Shank showed that syntax alone would not be enough and semantics would need to be included. Semantics and syntax were integrated in the 80s but it became apparent that pragmatics (intention) is also needed. While progress is being made, NLP and success in computation linguistics is still limited to small dictionaries and very complicated processing.

Further evidence of the limitations of current NLP systems is presented in (Lehnert & Sundheim, 1991). This article describes the winning system in a call made by the Naval Ocean Systems Center for a performance evaluation of existing text analysis systems. The systems were expected to process "essentially undoctored news articles about Latin American terrorism." Fifteen systems responded to the call. Scoring was based on standard recall, precision, overgeneration, and fallout measures. Only two

systems exhibited a recall score over 40 percent with precision over 60 percent. The major downfall of most systems was lack of good discourse analysis capabilities.

The lexical analysis associated with NLP today still has far to go before generalizable text processing is achievable. The "noise" found in EBS comments (typographical errors, poor sentence structure, unique vocabulary, etc.) combined with the unpredictable topics of discussion make the implementation of natural language understanding techniques impractical for concept creation.

3.2.2 Cluster Analysis

Cluster analysis techniques have been primarily used in statistics. The approach taken is typically serial in that input attributes are processed serially. Neural networks have provided support for processing input attributes in parallel using different mathematical functions and algorithms. These new parallel approaches combined with higher performance workstation computing power provide tools which allow testing of cluster analysis techniques that were previously impossible.

3.2.2.1 Serial Approaches

(Rasmussen, 1992) defines cluster analysis as "a statistical technique used to generate a category structure which fits a set of observations. The groups which are formed should have a high degree of association between members of the same group

and a low degree between members of different groups." The author points out that cluster analysis is a technique for multivariate analysis that has application in many fields. Thus, a number of software packages support various methods and algorithms (SAS, SPSSX, BMDP, CLUSTAN, CLUSTAR/CLUSTID). Typical problems that occur with using available software packages include: selecting the attributes on which items are to be clustered and their representation; selecting an appropriate clustering method and similarity measure from those available; assessing the validity of the result obtained; if the collection to be clustered is a dynamic one, the requirements for update must be considered; and if the aim is to use the clustered collection as the basis for information retrieval, a method for searching the clusters or cluster hierarchy must be selected. Measures of similarity between clusters include Dice coefficient, Jaccard coefficient, and cosine coefficient. Clustering algorithms divide into two major categories: non-hierarchical methods and hierarchical methods. Non-hierarchical algorithms include Single Pass, and Reallocation. Hierarchical algorithms include the Single Link method (Van Rijsbergen algorithm, SLINK algorithm, and Minimal spanning tree algorithms); Complete Link Method (Defray's CLINK algorithm and Voorhees algorithm); Group Average Link Method (Voorhees again); and Ward's Method (reciprocal nearest neighbor algorithm). For a complete review see (Everitt, 1980).

Automatic document classification involves determining a document representation structure and methods for determining similarities between the documents. The hierarchical clustering of documents can be carried out either *divisively* or *agglomeratively* (Salton, 1989 p. 329). Divisive clustering breaks one complete cluster down into smaller pieces. In agglomerative clustering "individual item similarities are used as a starting point and a gluing operation collects similar items, or groups, into larger groups (Salton, 1989)." Salton describes a 4-step program for agglomerative clustering that he adapted from (Griffiths, Robinson, & Willett, 1984):

- 1.) Compute all pairwise document-document similarity coefficients ($N(N-1)/2$ coefficients);
- 2.) Place each of N documents into a class of its own;
- 3.) Form a new cluster by combining the most similar pair of current clusters i and j ; calculate the entries in the row corresponding to the new cluster $i + j$;
- 4.) Repeat step 3 if the number of clusters left is greater than 1.

The principle structure for determining document similarity is the term-document matrix where each row of the matrix represents a document and each column represents a term. Each element of the matrix is a Boolean value indicating True if the given term is contained within the document and False if it is not. Thus, each

document can be represented by a vector and document collections can thus be represented in a vector-space model.

Robert Stepp (Stepp, 1987) describes conceptual clustering as the new frontier that arose from Michalski's introduction of the concept in 1980. Algorithms for clustering involve co-occurrences of feature values, discovering conjunctive features among the attributes rather than variations in the value taken by a single attribute, and clumping concepts based upon most commonly occurring relations in the data. Stepp takes two views of conceptual clustering -- as concept sorting and as concept discovery. Two stages occur in conceptual clustering: an aggregation phase and a characterization phase. The aggregation phase produces the groupings and the characterization phase attempts to assign some meaningful interpretation of the groupings. "That the clustering and characterization phases are not independent ... is precisely one of the facets that distinguishes conceptual clustering from 'regular' (statistical) clustering. (Stepp, 1987)"

Classes of similar objects are basically found by doing pairwise comparisons among all of the data elements. The above clustering algorithms are primarily serial in that pairwise comparisons are made one at a time. A newer approach for addressing clustering and classification problems is based on the connectionist approach, or neural network computing. Algorithms based upon neural networks are parallel approaches in

that multiple connections among the nodes allow for independent, parallel comparisons.

3.2.2.2 Parallel Approaches - Neural Network Clustering

Neural networks are patterned after the biological ganglia and synapses of the nervous system. The concept is not new -- McCulloch and Pitts suggested the description of a neuron as a *logical threshold limit* in 1943 (McCulloch & Pitts, 1943). The essential element of the neural network is the neuron. The typical neuron receives a set of input signals, x_i , each of which is multiplied by a weighting factor, w_j . All of the weighted input signals are added up to produce n , the net input to the neuron. An activation function, F , transforms the net input of the neuron into an output signal which is transmitted to other neurons (Dalton & Deshmane, 1991). Adjustments of the weighting factors and increase or decrease neuron output based upon their values being positive or negative, respectively.

Also, many types of activation functions can be used. For example, a linear activation function simply multiplies the sum of the input weights by some constant and outputs this new value. A threshold function will set the output of the neuron equal to a prescribed value (say, 1) if the net input exceeds a given threshold; otherwise 0 is output. A commonly used function is the sigmoid function, $F(n) = (1/(1+e^{-n}))$. This function is highly nonlinear, continuous, and continuously

differentiable. These characteristics make it very useful in making adjustments for neural network learning (Dalton & Deshmane, 1991).

The adjustments of the weights of the nodes of the neural network enable the total network to "learn" in that a neural network's performance can be adjusted to fit a known set of data characteristics. In supervised learning, a set of training examples are presented, one by one, to the network. The network then calculates outputs based on its current input. The resulting output is then compared with a desired output for that particular input example. The network weights are then adjusted to reduce the error. In unsupervised learning, network models are first presented with an input vector from the set of possible network inputs. The network learning rule adjusts the weights so that input examples are grouped into classes based on their statistical properties (Dalton & Deshmane, 1991) (Rumelhart, 1986).

A variety of network topologies can be used. Neurons can be layered with complete connections between each neuron of one layer and every neuron of the other layers. The network may be a single layer with connections among all of the neurons of the single layer such as that found in the Hopfield neural network (Hopfield, 1982).

(Lippmann, 1987) provides an overview of the various types of neural network topologies and the types of data that might be classified with them. He also classifies them with respect to supervised or unsupervised.

Concept classification of electronic brainstorming comments is an unsupervised learning approach. We treat each brainstorming comment as a document. Thus, representing the comments for classification using neural networks suggested a review of the information retrieval literature for suggestions on how to represent the comments and what neural network algorithms might be useful.

(Doszkocs, Reggia, & Lin, 1990) provide an overview of connectionist models in information retrieval including artificial neural networks, spreading activation models, associative networks, and parallel distributed processing. (Belew, 1989) developed a three-layer neural network of authors, index terms, and documents in a system called AIR. Relevance feedback from its users changed the representation of authors, index terms, and documents over time. (Kwok, 1989) used a modified Hebbian learning rule in a similar three-layer network of queries, index terms, and documents.

(MacLeod & Robertson, 1991) presents a two-layer neural network and an algorithm named the "MacLeod algorithm" that is used for document clustering. This algorithm is an unsupervised approach that performs clustering by a form of

partitioning. Inputs are binary-valued vectors representing documents. The algorithm uses two similarity measures in order to allow proof of algorithmic behavior, cluster stability, and characterization of final clusters. For each input vector, the goal is to find the closest cluster by the first similarity measure that is also sufficiently close by the second. The algorithm is multi-pass in that documents are repeatedly read into the network until two consecutive passes produce identical classifications for each document (convergence). Only those clusters that are successful in classifying one or more documents during the last pass are active. Inactive clusters do not take part in any subsequent retrieval operations.

(Chen, Hsu, Orwig, Hoopes, & Nunamaker, 1994) describe an algorithm for concept classification of electronic brainstorming comments that combines automatic indexing of the electronic brainstorming comments, term co-occurrence concept space generation, and a Hopfield neural network classifier. Results of experiments comparing the output of their algorithm with human experts and novices found that the Hopfield classifier performed as well as the novices but two human experts outperformed the novices and Hopfield classifier significantly. Reasons given for the performance differences centered on the ability of the experts to associate terms more appropriately.

A review of (Lippmann, 1987) in view of the almost continuous nature of the weights assigned to term pairs in the co-occurrence concept space and the unsupervised learning nature of the task suggested the Kohonen Self-Organizing Map as a possible algorithm for use in classification of electronic brainstorming comments.

Kohonen bases his neural network on the associative neural properties of the brain (Kohonen, 1989). This network contains two layers of nodes -- an input layer and a mapping layer in the shape of a two-dimensional grid (Caudill, 1993) (Hiotis, 1993). The input layer acts as a distribution layer. The number of nodes in the input layer is equal to the number of features or attributes associated with the input. Each node of the mapping layer also has the same number of features as there are input nodes. Thus, the input layer and each node of the mapping layer can be represented as a vector of a dimension equal to the number of features of the input. The network is fully connected in that every mapping node is connected to every input node. The mapping nodes may be initialized with random numbers. Each input gets compared with each node on the mapping grid. The "closest" mapping grid is defined as that with the smallest Euclidean distance between the mapping node vector and the input vector. The input thus maps to the closest mapping grid. However, the value of the mapping grid vector is adjusted according to the difference between the input and the current state of the mapping grid neuron. In addition, all of the neighboring nodes of the winning neuron are adjusted. In this way, the multi-dimensional input nodes get

mapped to a two-dimension output grid. After all of the input is processed, the result should be a spatial organization of the input data thus organized into clusters of similarity.

(Ritter & Kohonen, 1989) apply the Kohonen SOM to text and make two attempts to detect the logical similarity between words from the statistics of their contexts. The first approach represents context of terms as a set of attribute values that occur in conjunction with the words. The second approach defines context by the sequences in which words occur without consideration of any associated attributes. They analyze simple verbal statements consisting of nouns, verbs, and adverbs and suggest that "such phrases or clauses involve some of the abstractions that appear in thinking, namely, the most common categories, into which the words are then automatically grouped. (Ritter & Kohonen, 1989)" They argue that a similar process may be at work in the brain. Typical self-organizing maps reflect metric distance relations between patterned representation vectors. Ritter and Kohonen associate this with lower levels of perception. "However, much of higher-level processing, in particular *language* and *reasoning*, seems to rest on the processing of *discrete symbols*. Hence we must understand how the brain might form meaningful representations of symbolic entities. (Ritter & Kohonen, 1989 p. 246)"

Ritter and Kohonen argue that it is in the very nature of a symbol that its meaning is dissociated from its encoding and, hence, logical relatedness between different symbols will not be detectable from their encodings. To map symbols topographically one must present the symbol "in due context, i.e. in conjunction with all or part of the attribute values of the item it encodes, or with other, correlating symbols. (Ritter & Kohonen, 1989 p. 247)." The example they give uses sixteen animal types (dove, hen, fox, tiger, cow, etc.) and thirteen attributes (size, number of legs, hunts, swims, etc.). Input of these attributes for each of the animal types into the SOM produces an organization of the animals on a map where animals are spatially ordered into essential "family-relationships."

Ritter and Kohonen's second example involves three-word phrases of nouns, verbs, and adverbs. Their stated purpose is "only to demonstrate the *potential* of a self-organizing process to form abstract maps (Ritter & Kohonen, 1989 p. 250)" and their concerns are biological in that they seem more involved with whether these self-organizing maps directly correlate to semantic brain maps rather than the utility of the map itself. However, they do indicate that words, as symbols, can be used with the SOM with context.

Risto Miikkulainen (Miikkulainen, 1993) developed DISCERN (DIstributed SCRipt processing and Episodic memoRy Network) as his dissertation project. DISCERN is a

prototype of a subsymbolic natural language processing system that shows how script instantiation and inferences can be learned from examples by a parallel distributed processing architecture that is based upon a Kohonen Self-Organizing Map. However, DISCERN does not implement the entire computational theory or even the full complexity of symbolic script-processing systems. Miikkulainen provides an interesting discussion concerning expansion of classification into a hierarchy of layers of abstraction whereby classification occurs between each level of abstraction. In the case of scripts, a given script is at the highest level supported by tracks which are further supported by role bindings or instantiation levels. He suggests using Kohonen feature maps as representations of these hierarchical levels.

In support of using Kohonen for textual document classification, (Lin, Soergei, & Marchionini, 1991) use the Kohonen map for classifying documents for information retrieval. Documents are represented as vectors of binary values. Each coordinate of the vector represents a specific term or term phrase with the value set to '1' if the term or term phrase is found within the document and '0' otherwise. After several passes through the input file the Kohonen layer is trained. The resulting map provides an intuitively-appealing organization of the input data. The documents are classified according to their textual content and the method described in this article provide a blueprint for using the Kohonen self-organizing map in an actual application.

Lin et. al. demonstrated the feasibility of using the Kohonen algorithm for classification of textual documents. In addition, since associated terms were juxtaposed on the map, the algorithm provided potential for suggesting association of terms in a manner that might approach the experts in the Chen et. al. study. Further, the graphical output was appealing for use with typical electronic meeting participants (non-computer people). Therefore, we decided to use this Kohonen approach to analysis of the EBS comments using automatic indexing techniques to identify the meaningful terms. Consistent with (Chen et.al., 1994) we used evaluation techniques from the Information Sciences to validate our output.

CHAPTER 4: RESEARCH DESIGN AND PREVIOUS WORK

The system development research process described in (Nunamaker, Chen, & Purdin, 1991) (Figure 12) provides the methodology used in this research. This research spans two iterations of this methodology. During the first iteration, a conceptual framework was constructed, a system architecture was developed, the

Figure 12 -- A process for Systems Development Research

analysis and design of a system was completed, and a prototype system was built. A

summary of this iteration is described below. A complete description of the development and evaluation of this prototype system was reported in (Chen, Hsu, Orwig, Hoopes, & Nunamaker, 1994). The research question during this first iteration was "Can a Hopfield Neural Network be used to classify electronic brainstorming comments in a manner that equals or exceeds the performance of human classifiers?" Results of this first prototype indicated that the algorithm did perform as well as two of four human facilitators.

However, two of the facilitators were judged by all four facilitators to classify EBS comments in a manner that was significantly better than the Hopfield system and the other two facilitators. Even the list produced by the group that created the EBS data was not judged to be as good as those made by the two expert facilitators.

Evaluation comments by the facilitators concerning items on the lists indicated that the two lists were superior due to the high degree of association of the terms that formed the concepts of the lists. Figure 13 is an example of the concept space of co-occurrence weights created as output of the cluster analysis in (Chen, et. al., 1994). Observation of the weights that were assigned to terms in the study by the Hopfield algorithm indicated an almost continuous relationship among the weights of the data. Another review of the neural network literature pointed to a suggestion of using the Kohonen Self-Organizing Map (SOM) due to the high degree of noise in the textual

```

PRODUCT : PRODUCT: 1.0000
PRODUCT : DESIGN: 0.1869
PRODUCT : DEVELOPMENT: 0.1107

DESIGN : DESIGN: 1.0000
DESIGN : PRODUCT: 0.3460

MARKET : MARKET: 1.0000
MARKET : PRODUCT: 0.3373
MARKET : DESIGN: 0.1887

TIME : TIME: 1.0000
TIME : DESIGN: 0.1242
TIME : PRODUCT: 0.1179
TIME : SCHEDULE: 0.1023

DEVELOPMENT : DEVELOPMENT: 1.0000
DEVELOPMENT : PRODUCT: 0.4739
DEVELOPMENT : PRODUCT DEVELOPMENT: 0.2205
DEVELOPMENT : PROJECT: 0.1339
.
.
.

```

Figure 13 -- Cluster Analysis Output (Chen, 1994)

data and almost continuous nature of the weights that were assigned to the terms. The Kohonen SOM also provides associative relationships among the terms of the map such that related terms are co-located. This property was deemed to be desirable as a potential for providing a simpler user interface.

4.1 RESEARCH QUESTIONS

Two research questions are associated with this new iteration: Can the Kohonen SOM perform as well or better than the Hopfield network prototype in suggesting a

list of concepts? Also, does the Kohonen SOM associate terms as well as a human expert?

Thus, this research followed the iterative nature of the System Development Research Process and provided another iteration of the methodology.

4.2 HOW PEOPLE CLASSIFY EBS COMMENTS

Facilitators were video-taped during their processes of classifying electronic brainstorming output in (Chen, 1994). In addition, notes were taken of observations of the processes that the facilitators used in examining the electronic brainstorming output. Each facilitator made a single, serial pass through the comments.

4.2.1 A Serial Comparison of Each Comment

Each facilitator went through the comments serially, constantly comparing the most recent comment to the list of concepts created up until the current comment with optional changes to the existing concept list. This serial nature of comparison of the current input with the existing map contrasts with the Hopfield algorithmic approach which would be manifested in human behavior by reading one comment and then looking through the rest of the comments for related comments.

4.2.1.1 Human Process. Four facilitators were given two sets of EBS comments and asked to create a list of topics that they felt, in view of the EBS question, would be appropriate for a group to vote on. They were asked to provide their criteria of a good topic list in advance of creating the lists.

Facilitators shared the same basic criteria: a list of topics must cover the issues discussed during the EBS session and be appropriate for the EBS question; each topic must be at about the same level of abstraction; each topic must be unique with respect to the other topics on the list; and the ideal list contains less than twenty items.

Each facilitator familiarized themselves with the EBS question of the comment set that they were working on and browsed the first few comments. Eventually they would post the first topic item to their blank list. They would then proceed through the set of EBS comment in order and complete the list in one pass.

While proceeding through the comment file they would read a comment and determine its relevance to the EBS question. If it was relevant, they would compare the subject matter of the comment against the current list of topics. If it was a new topic and "worthy" of being added to the list they would add it. "Worthiness" was defined by the facilitators as a topic being at the same level of abstraction as those items on the list.

Worthiness also included whether there was more than comment concerning the same topic. Facilitators seemed to also keep a frequency counter going in their heads that would be used to elevate new topic suggestions to topic-list status if more than one comment supported it.

Facilitators would sometimes adjust the topic list to accommodate a different point of view that a particular comment would have given them or they would adjust the topics to balance them for equivalent levels of abstraction.

Facilitators typically took fifty minutes to an hour to create a list of topics.

4.2.1.2 General Kohonen Algorithm. The Kohonen SOM works much in the same way as facilitators in processing the EBS file: serially going through the data, comparing each comment against the current state of the map, determining which node is closest, and making adjustments to not only the "winning" closest node but also other nodes surrounding the winning node.

The comparison, in both cases, is looking for topics which are most similar to the input currently under question.

Differences, however, include the fact that Kohonen always has a "winning" node. It does not add new topics. Also, the Kohonen algorithm adjusts the weights of each neighbor of the winning node every time.

4.2.2 An Example: Human Approach to Real-time Classification

Subject Facilitators of the cognitive study performed the classification as a stand-alone process. They could take all of the time that they wanted and could remain focused. In actual sessions, the facilitator is leading the group of participants and tries to off load as much of the classification process as possible onto the participants. Different techniques that facilitators use for convergence with a group are described above.

This example describes an actual session in which the facilitator created the list of keywords during the time that the group was creating the brainstorming output.

A group of approximately thirty people representing several organizations met during a conference. Since the conference involved electronic meeting systems, these participants all had an interest in EMS and in computer-supported collaborative work. The desired outcome of the session was a rank-ordered list of collaborative technology problems to be solved within the next five years. This EBS question was posed to the group for electronic discussion.

The facilitator could read the comments as participants finished them and pressed a key to read other comments. After participants had entered several comments the facilitator began using the keyword option in EBS. The facilitator was an expert and taught courses in EMS and Executive Information Systems. Thus, the facilitator had more than adequate knowledge of the subject domain. The intent of using the keyword feature was to save time in the convergence process by giving the participants a list with which to start analysis of the EBS comments rather than start them with an empty list. After about one-half hour of EBS 202 comments were generated. A sample of the comments is displayed in Figure 6.

The list that was generated by the facilitator is shown in Figure 14. When the facilitator identifies keywords during the EBS session the program assembles all of the comments that contain the entered keyword and connects them to that topic. In this way, comments which discuss any given keyword will be attached to support it and serve as reference for the group to validate the topic as important to the EBS question. Because keywords are not always "semantically rich," the group can also decide to change to wording in the convergence stage to something more meaningful. An example of a semantically-deficient term from Figure 14 would be item number 5, "voice." This term was used to collect all of the comments that contained "voice recognition," "data/voice communication," etc. The group would most likely change the topic name or add other terms to make item number 5 more meaningful.

What are the most important information technology problems with respect to Collaborative Systems to be solved over the next five years?

1. video / projection
2. network / bandwidth
3. multimedia / hypertext / multi-media
4. group memory / project memory / repository
5. voice
6. culture / style
7. language
8. standards
9. distributed / distance issues / distance / different place
10. facilitation
11. research methodologies
12. cost / money
13. team
14. reward
15. integration
16. social / societal / society
17. performance
18. virtual
19. education / train / learn / teach
20. human / people / user / individual / interpersonal

Figure 14 -- Facilitator-Generated Topic List

The goal of a classification algorithm is to generate a list that at least is as good at that depicted in Figure 14.

4.3 HOPFIELD NET CLASSIFICATION - PREVIOUS WORK

A summary of the Hopfield Net Algorithm that was used in the first iteration of research is described below. The portions of Hopfield Net Algorithm that pertain to

this research include a portion of the design of the algorithm as applied to the EBS comments, participation in the development of the prototype, and evaluation of the prototype by performing an experiment with four facilitator-subjects.

The Hopfield Neural Network (NN) approach to classification of EBS comments is documented in (Chen, Hsu, Orwig, Hoopes, & Nunamaker, 1994). The algorithm consists of three major stages: Automatic Indexing of the EBS Comments, Concept Space Generation, and Classification using the Hopfield NN.

4.3.1 Automatic Indexing of EBS Comments

We follow automatic indexing techniques typically used in the Information Sciences by treating each comment in the EBS file as a document. However, each EBS comment is limited to a maximum of five lines of text. Examination of actual EBS comments, however, reveal that a majority of the ideas proposed are expressed in simple terms or phrases. In all cases, each comment (document) was indexed using four steps: word identification, stop-wording, stemming, and term-phrase formation. Punctuation symbols were ignored and words were stored along with the comment number, paragraph number, sentence number, and the position of the word within the sentence. We developed a list of non-semantic bearing words and removed any terms matching any of the terms found in this list from the comments. Stemming reduces the variations of terms due to pluralization or other suffix addition by reducing terms

```

1.1 1 1 2 1 DEVELOPMENT
1.1 1 1 4 1 STANDARD
1.1 1 1 5 1 USER
1.1 1 1 5 2 USER INTERFACES
1.1 1 1 6 1 INTERFACES
1.1 1 1 8 1 REOMTE
1.1 1 1 11 1 INFO
1.1 1 1 13 1 COMMUNICATION
1.2 1 1 3 1 XYZ
1.2 1 1 3 2 XYZ NOTES
...
30.7 1 1 3 1 MATTER
30.7 1 1 10 1 LIMITED
30.7 1 1 17 1 LIMITED
30.7 1 2 19 1 GOD
30.7 1 2 21 1 UNANSWERED
30.7 1 2 21 2 UNANSWERED PRAYERS
30.7 1 2 22 1 PRAYERS

```

└─ number of words in the term
 └─ position number of word in sentence
 └─ sentence number of paragraph
 └─ paragraph number
 └─ EBS comment number

Figure 15 -- Sample Automatic Indexing Output

to a root word. Finally, 2 and 3 adjacent terms were combined into term phrases and treated as one term.

Automatic indexing produces an index in a format depicted in Figure 15. Each of the comments is uniquely numbered. In this case, the EBS program produced thirty files. Comment 1.1 is the first comment of the first file. The first eight terms indexed in Figure 15 all result from the very first comment. The first two terms of the

second comment of the first file (1.2) are also shown. Note that automatic indexing output includes term phrases up to three terms in length.

4.3.2 Concept Space Generation

After indexes had been assigned to each comment, we identify the co-occurrence pattern of indexes which appear in all comments. A similarity measure similar to the popular Cosine function (Salton, 1989) is adopted. Using the weights computed from the similarity measure, our system generates a network, or concept space, to represent the weighted relationships between indexes (concepts) in the documents. We summarize the procedure below.

4.3.2.1 Term Frequency and Document Frequency Computation

We first compute the *term frequency*, tf_{ij} , which represents the number of occurrences of term j in document i and the *document frequency*, df_j , which represents the number of documents in a collection of n documents in which term j occurs.

4.3.2.2 Determining Threshold Through Information Loss Analysis

We then perform an information loss analysis in order to determine the threshold for removing uncommon terms (e.g., typos and unique abbreviations) and for generating a set of consensus terms for further analysis. Only indexes that pass the threshold are considered in the Hopfield classification process. An EBS comment was

considered "lost" if all its indexes appeared less often than a specified document frequency threshold. These "lost" comments, however, can still be retrieved through index search or browsing. We found that selecting a threshold where only 10% of the EBS comments were "lost" reduced the number of terms from thousands in the index to just over one hundred in most cases. This greatly influenced program running time due to the polynomial ($O(n^2)$) classification procedure where n is the number of terms.

After applying the threshold, the remaining terms are ranked according to their decreasing order of document frequency (Figure 16).

4.3.2.3 Combining Weights

We then compute the combined weight of term j in document i , d_{ij} , based on the product of *term frequency* and *document frequency* as follows:

$$d_{ij} = tf_{ij} \times \log df_j$$

4.3.2.4 Concept Space Generation

Based on the asymmetric *Cluster Function* shown below, we generate a network-like concept space of terms and their weighted relationships.

0	TECHNOLOGY
1	MEETINGS
2	COLLABORATIVE SYSTEMS
3	SUPPORT
4	TOOLS
5	COLLABORATIVE
6	SYSTEMS
7	MEETING
8	DISTRIBUTED
9	PEOPLE
10	ENVIRONMENTS
11	TECHNOLOGIES
12	ENVIRONMENT
13	INFORMATION
14	VIDEO
15	NETWORKS
16	ISSUES
17	NETWORK
18	LINEAR THREAD MEETING
19	HARDWARE
20	DISTRIBUTED MEETINGS
21	PROCESSES
22	CULTURAL
...	
181	CHANGE
182	WORLD
183	COMMON SPOKEN LANGUAGE
184	DIFFERENCES
185	REMOTE COLLABORATIVE MEETINGS
186	WINDOWS
187	REMOTE SCREEN SHARING
188	MODE
189	RELIABILITY
190	LIMITED

Figure 16 -- Terms Ranked according to decreasing document frequency

$$ClusterWeight(T_j, T_k) = \frac{\sum_{i=1}^n d_{ijk}}{\sum_{i=1}^n d_{ij}}$$

$$ClusterWeight(T_k, T_j) = \frac{\sum_{i=1}^n d_{ijk}}{\sum_{i=1}^n d_{ik}}$$

These two equations compute the similarity weights from term T_j to term T_k (the first equation) and from term T_k to term T_j (the second equation), where d_{ij} represents the combined weight of descriptor T_j in document i , and d_{ijk} represents the combined weight of both descriptors T_j and T_k in document i . d_{ijk} is computed by:

$$d_{ijk} = tf_{ijk} \times \log df_{jk}$$

where tf_{ijk} represents the number of occurrences of both term j and term k in document i and df_{jk} represents the number of documents in a collection of n documents in which both term j and term k occur.

4.3.3 Hopfield Net Classification

The final output from the Concept Space Generation is a matrix of descriptors and their weighted relationships, akin to a neural network of nodes and weighted links. Our system adopts a variant of the Hopfield network activation procedure to identify *clusters* of relevant descriptors in the concept space through their weighted links.

Each term in the concept space is treated as a neuron and the asymmetric weight between any two terms is taken as the unidirectional, weighted connection between neurons. Using each remaining term after *information loss analysis* as the individual input pattern, the Hopfield algorithm activates its neighbors, combines weights from all associated neighbors, and repeats this process until the output pattern converges. The resulting output reveals all other concepts that are semantically relevant to the input term. Repeating this process for all the terms according to their decreasing occurrence frequencies, we are able to identify the underlying, related concepts in the EBS comments. A sketch of the *Hopfield net concept classification* procedure follows:

4.3.3.1 Assigning Connection Weights

The training phase of the Hopfield net was completed when the concept space was created. t_{ij} represents the synaptic weight from node i to node j .

4.3.3.2 Initialization with Unknown Input Pattern

$$\mu_i(0) = x_i, \quad 0 \leq i \leq n-1$$

$\mu_i(t)$ is the output of node i at time t . x_i , which has a value between 0 and 1, indicates the input pattern for node i . At time 0, the starting input node is assigned 1. Each term is used individually as an input node for activation, i.e., at time 0 only one

term receives the value 1 and all other terms receive 0, and we repeat this initialization and the following activation process n times.

4.3.3.3 Activation and Iteration

$$\text{for term } j, 0 \leq j \leq n-1, \mu_j(t+1) = f_s \left[\sum_{i=0}^{n-1} t_{ij} \mu_i(t) \right]$$

where f_s is the continuous sigmoid transformation function (Knight, 1990) (Dalton & Deshmane, 1991) as shown below.

$$f_s(net_j) = \frac{1}{1 + \exp \left[\frac{-(net_j - \theta_j)}{\theta_0} \right]}$$

where $net_j = \sum_{i=0}^{n-1} t_{ij} \mu_i(t)$, θ_j serves as a threshold or bias and θ_0 is used to modify the shape of the sigmoid function.

4.3.3.4 Convergence

The above process is repeated until there is no change in terms of output between two iterations, which is accomplished by checking:

$$\sum_{j=0}^{n-1} [\mu_j(t+1) - \mu_j(t)]^2 \leq \epsilon$$

where ϵ is the maximal allowable error (a small number). The final output represents the set of terms relevant to the starting term.

A diagram and summary of the Hopfield Algorithm is shown in Figure 17.

Figure 17 -- Hopfield Architecture and Algorithm

4.3.4 Hopfield Classification Example

The Hopfield Classification Algorithm was being demonstrated to the participants attending the meeting described in 4.2.2. As part of the demonstration, the EBS comments generated by the participants was input into the Hopfield Algorithm. The

What are the most important information technology problems with respect to Collaborative Systems to be solved over the next five years?

1. technology / dealing / cultural differences
2. collaborative systems
3. meetings / distributed / human
4. linear thread meeting
5. ai / data / amounts
6. environments / virtual
7. voice recognition
8. tools / culture
9. people
10. technologies
11. information / ability
12. video / desktop
13. networks
14. issues
15. hardware
16. distributed meetings
17. training
18. groups
19. applications
20. distributed environment

Figure 18 -- Hopfield Classifier Generated List

output is presented in Figure 18.

CHAPTER 5: KOHONEN CLASSIFICATION FOR EBS COMMENTS

This research is primarily focused on the second iteration of the systems development methodology whereby the Hopfield Algorithm was replaced by the Kohonen Self-Organizing Map (SOM) prototype.

The Kohonen SOM was developed by Tuevo Kohonen at the University of Helsinki (Kohonen, 1989). His ideas for the self-organizing map come out of the

Figure 19 -- Kohonen Self-Organizing Map Architecture

biological sciences and the physical modeling of memory.

"One particular difficulty in the physical modelling of memory arises from the fact that the human mind, as a result of a long intellectual evolution, is able to deal with *concepts* that appear to be distinct and unique items, almost like material objects. The degree of specificity connected with the concepts is so high that very complex abstract structures of knowledge have been formed into our thinking. This development has led to the creation of languages and especially of the written language which obeys its own strict rules. It has often been stipulated that any concrete models of memory processes ought to reflect similar complexity and perfection; therefore the modelling task does not seem very easy or tempting. (Kohonen, 1989 p. 1)"

However, Kohonen goes on to indicate that it seems possible that representations even for abstract items may be formed in certain self-organizing processes.

The fundamental architecture and algorithm steps for the Kohonen SOM are shown in Figure 19. The architecture is such that there is only one input layer and one map layer of neurons. The map layer neurons form a grid of arbitrary dimension. Each node of the input layer is fully connected to every node on the Kohonen layer. Each node on both layers is a vector of dimension n where n equals the number of terms. Each node of the input layer represents one comment. Each node (comment) is thus represented by a vector with a '1' for terms represented in the comment and '0' otherwise. The connections between the input layer and the Kohonen layer represent the weights.

Each node on the Kohonen layer is associated with a "neighborhood." A node's neighborhood consists of the neurons surrounding it. The size of the neighborhood changes as input data is processed. The shape of the neighborhood is also changeable.

5.1 KOHONEN CLASSIFICATION ALGORITHM

The algorithm for classifying EBS comments using the Kohonen SOM uses the same automatic indexing process as the Hopfield algorithm. Output of the indexing process is converted into a form compatible with the Kohonen input. Program listings for the software that performs this conversion is attached as Appendix B.

Each comment of the EBS is represented by a vector of 1's and 0's. The number of items in each vector is equal to the number of terms in the document frequency list and each vector position corresponds to one term in the frequency list. The Kohonen algorithm initializes the Kohonen layer and processes the input file in a serial manner.

5.1.1 Initialize the Kohonen Layer

Initialize the weights connecting the inputs to the Kohonen layer to small random values and initialize the neighborhood type and size.

5.1.2 Process Input

Present an input and compute the difference/similarity between this input and the current map. This similarity measure is distance and is computed by

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2$$

where $x_i(t)$ is the input to node i at time t and $w_{ij}(t)$ is the weight from input node i to output node j at time t .

The node on the grid with the minimum d_j is the "winning" node. Use this node as the center of the neighborhood.

5.1.3 Adjust Neighbors

With the "winning" node identified (say j^*), adjust the weights for the minimum node and all of the nodes in its defined neighborhood. The new weights are computed as

$$w_{ij^*}(t+1) = w_{ij^*}(t) + \eta(t) (x_i(t) - w_{ij^*}(t))$$

for j^* , a node in the neighborhood and where $0 \leq i \leq N - 1$. The gain term, $\eta(t)$ where $0 < \eta(t) < 1$, decreases in time. See (Lippmann, 1987) (Hiotis, 1993) (Caudill, 1993) and (Kohonen, 1989) for more details of the algorithm.

5.2 KOHONEN CLASSIFICATION EXAMPLE

The same EBS output data as used in Figure 6 is used in this example. Also, the same automatic indexing software used to produce the index shown in Figure 15 and Figure 16 were used for preparation for the Kohonen Classifier. A summary of the processing of EBS comments for Kohonen follows. For a more complete description of the process, see Appendix D.

5.2.1 Convert Index and Frequency List

Automatic indexing is performed on the EBS comment file in a manner identical with that performed with the Hopfield algorithm. Output of the automatic indexing contains an index for the terms of the comments as represented in Figure 15 and a list of terms in decreasing order of frequency as represented in Figure 16. These two files are used as input to create a Kohonen input file (input.dat) which is a file of 212 vectors where 212 is the number of comments contained in the electronic brainstorming file. Each vector is of dimension 190 where 190 is the number of terms contained in the file represented by Figure 16. The first value of each vector corresponds to the first term on the file represented by Figure 16. The second value of each vector corresponds to the second term (and so on). The value of each vector value is '1' if the term that it represents is contained within the comment and '0' otherwise.

5.2.2 Create Unit Vector Input

In addition to the comment data file, a file of vectors of 190 dimensions is also needed by the Kohonen program to locate terms on the Kohonen Map and for determining the map regions. This file (unitvect.dat) contains 190 vectors with the first vector having a '1' in position one and '0' in all other positions. Vector two has a '0' in position one, a '1' in position two, and '0' in all remaining positions. Vector 190 has all '0' in positions 1 through 189 and a '1' in position 190. Appendix C contains the short program that creates a Kohonen input file that is used for identifying locations of terms on the Kohonen map.

5.2.3 Create the Map

The Kohonen program consists of four modules: map initialization, map training, quantization error computation, and map visualization. A more detailed explanation of the steps is contained within Appendix D.

5.2.3.1 Map Initialization

Several parameters are passed to the initialization program: the size of the map to create in terms of the number of grid points on the horizontal and vertical dimensions, the topology of the neighborhood, and the type of neighborhood adjustment to use. We used a 20 by 10 grid map for our example based upon what would fit on an output screen. The neighborhood surrounding each node on the grid can be viewed as

a rectangular area or hexagonal area. The rectangular area considers eight neighboring grid nodes surrounding each interior map node. We used a hexagonal area which considers six surrounding nodes to be a node's immediate neighborhood. Finally, two types of neighborhood weight adjustments can be made: gaussian or bubble. The gaussian function adjusts the weights of each neighbor according to a value computed from a gain term in combination with the immediate size of the neighborhood. We used the bubble adjustment method which is simply an adjustment of the weights of neighboring nodes based upon a simple calculation of the gain term.

The vector values for each node on the grid are initialized to small random values.

5.2.3.2 Map Training

The map training occurs in two phases: an ordering phase during which the vectors of the map take on rough values followed by a fine-tuning phase. Three parameters that are passed to the program determine whether the phase is an ordering phase or a fine-tuning phase: number of iterations, gain adjustment, and size of the neighborhood radius. The ordering phase was run with 1000 iterations of the input, an initial weight gain adjustment of .05, and a neighborhood size of 10. In contrast, the fine-tuning phase was run with 10,000 iterations of the input, an initial weight gain adjustment of .02, and a neighborhood radius of 3. The ordering phase begins with a larger neighborhood and larger adjustment in weights being made to neighboring

nodes. The phase ends after 1,000 input nodes have been processed (recycling through the input file if necessary) or until the neighborhood size shrinks to 0 -- whichever comes first.

Output of this step is a file named `map.cod` which is a 10 by 20 grid of 190-dimension vector weights.

5.2.3.3 Quantization Error Computation

The original input data file is run once against the `map.cod` file. For each input input vector a map node is identified as a winning node and the Euclidean distance between the two vectors is computed. The output of the quantization error step contains the average error over all of the input vectors.

5.2.3.4 Map Visualization

Visualization consists of running an input file against the map and reporting the map grid location that is closest in Euclidean distance to each input in (row, column) format as well as reporting on the size of the Euclidean distance (individual quantization error). Running the input data file against the `map.cod` file with the visualization program will report which map grid node is most closely aligned with each comment. In this way, comments can be plotted onto the map.

Similarly, the `unitvect.dat` file can be used as input to the visualization program thus reporting which map grid node is most closely aligned with each term (thus treating each term as a 1-word comment.)

To get the map regions, run the visualization program again but treat the `unitvect.dat` file as a map and use the `map.cod` file as input. This finds a winning node for each vector of the map grid from the `unitvect.dat` file. Thus, each node on the grid gets assigned one of the terms. Labeling each node on the map with the appropriate term resulted in the map shown in Figure 20.

5.2.4 Interpret the Output

The output of the visual programs are lists of locations on the grid for the comments and terms. Also, output contains the "value" of each cell in the grid relative to the terms list. The map produced by the Kohonen algorithm for the example is shown in Figure 20.

Terms found on Figure 20 seem to occupy regions of the map with very few regions being split by intervening terms ("cultural" and "collaborative", for example). Even when regions are split, the regions are located in the same general area of the map.

Figure 20 -- Kohonen Output

There also seems to be a general relationship between most of the terms that are adjacent on the map. "Collaborative" and "systems" are terms frequently used together as well as "distributed" and "meetings."

The size of the area seems to also bear some relationship to the importance of the topic. The largest amount of area is represented by terms that come directly from the brainstorming question, "What are the most important information *technology* problems with respect to *collaborative systems* to be solved over the next five years?".

Several questions arise in looking at the Kohonen output: Does the Kohonen neural network select terms any better than the Hopfield Algorithm? How does it compare with a human performing the same task? Does the size of the area bear any relationship to the importance or volume of the comments pertaining to topic? How relevant are the identified terms with respect to generating topics related to the EBS question? How strong is the degree of association among the terms that define adjacent regions of the map?

CHAPTER 6: EVALUATION OF KOHONEN CLASSIFICATION

6.1 EXPERIMENT DESCRIPTION

Two primary questions arise in observing the Kohonen output: How does a list of topics identified by the Kohonen SOM compare with the quality of the Hopfield and human lists? Does the co-location of regions signify a higher degree of association between the terms identifying those regions than regions that are separated on the map? To address questions arising from observation of the Kohonen output, we designed a three-stage experiment.

The first stage involved comparing the topic lists generated by a human expert, the Hopfield Algorithm, and the Kohonen SOM. This stage addresses list quality in comparison of the output of the human expert and Hopfield Algorithm.

The second question requires a finer unit of analysis. Since each topic or concept on the list is a term, term phrase, or multiple terms or term phrases we need to analyze those terms that comprise the concept. Verbal protocols from human analysis of expert facilitator lists indicated that the better list items contain terms which are relevant to the Electronic Brainstorming question and are better associated in forming the concept or topic.

Ideally, each concept (list item) should be semantically discrete with respect to the other items on the list. However, we did not test for this characteristic.

To address the issue of term relevance, we pooled all of the terms in all three lists, presented them to the subjects in alphabetical order, and asked the subjects to rate the relevance of the terms to the EBS question. Thus, the second stage measured the degree of relevance of the terms contained with the topic lists.

The third stage measured the relative degrees of association that existed between each of the terms on the three lists. This stage addresses the question concerning the adjacent areas of the Kohonen map and whether the apparent high degree of association actually exists between the areas. If so, the Kohonen SOM might prove more useful in supporting group convergence in suggesting conceptual clusters as regions in the map.

6.1.1 First Stage: List Comparison

Text output from an electronic brainstorming session from an actual electronic meeting was used. During the electronic brainstorming, the facilitator created a set of keywords for categorizing the comments pertaining to the question: "What are the most important information technology problems with respect to Collaborative Systems to be solved over the next five years?" The EBS output was also run through the

Table I -- Three Concept Lists

Facilitator	Hopfield	Kohonen
1. video / projection	1. technology / dealing /	1. technology
2. network / bandwidth	cultural differences	2. tools
3. multimedia / hypertext /	2. collaborative systems	3. meeting
multi-media	3. meetings / distributed /	4. support
4. group memory / project	human	5. collaborative
memory / repository	4. linear thread meeting	6. facilitator
5. voice	5. ai / data / amounts	7. matter
6. culture / style	6. environments / virtual	8. systems
7. language	7. voice recognition	9. people / issues /
8. standards	8. tools / culture	communication
9. distributed / distance	9. people	10. application /
issues / distance /	10. technologies	integration / groupware
different place	11. information / ability	11. recognition
10. facilitation	12. video / desktop	12. cultural
11. research	13. networks	13. video
methodologies	14. issues	14. hardware
12. cost / money	15. hardware	15. networks
13. team	16. distributed meetings	16. dealing
14. reward	17. training	17. users / group / reward
15. integration	18. groups	18. collaboration /
16. social / societal /	19. applications	seamless
society	20. distributed	19. memory / tool
17. performance	environment	20. training
18. virtual		21. virtual
19. education / train /		22. environment
learn / teach		23. notes
20. human / people / user		24. information / ability
/ individual / interpersonal		25. distributed
		26. standard
		27. bandwidth

Hopfield and Kohonen Algorithms. The concept lists created by the three methods are shown in Table I.

The terms comprising regions of the Kohonen Map were put into list form. Attached to these terms were any other terms that mapped into the same region through running a unit vector that represented each of the terms against the map.

Subjects of the experiment were given the text output of the electronic brainstorming session and the three lists -- each unidentified on a separate sheet of paper. Subjects read through the comments (optionally creating their own lists). They then ranked the three lists from most appropriate to least appropriate with respect to identifying types of information technology problems as addressed by their interpretation of the EBS question. Subjects were then asked to "correct" each list by deleting inappropriate terms and adding terms that were missing.

Subjects completed this stage in times that ranged from just under one hour to just over two hours.

6.1.2 Second Stage: Term Relevance

The terms on the three lists (Facilitator, Hopfield, and Kohonen) were pooled into one alphabetical list of 83 terms (Table II). Combinations of terms forming individual concepts (e.g., "video / projection") were split apart and treated as separate terms. Term phrases were retained (e.g., "distance issues" was treated as one term as was "distance"). Subjects were not allowed to re-visit the individual lists to determine the source of the terms.

Subjects were asked to rate the relevance of the terms with respect to whether they should be included within a concept list pertaining to the electronic brainstorming

Table II -- Pooled List of Identified Terms

1. ability	27. facilitation	55. project
2. ai	28. facilitator	memory
3. amounts	29. group memory	56. projection
4. application	30. group	57. recognition
5. applications	31. groups	58. repository
6. bandwidth	32. groupware	59. research
7. collaboration	33. hardware	methodologies
8. collaborative	34. human	60. reward
9. collaborative	35. hypertext	61. seamless
systems	36. individual	62. social
10. communication	37. information	63. societal
11. cost	38. integration	64. society
12. cultural	39. inter-	65. standard
13. cultural	personal	66. standards
differences	40. issues	67. style
14. culture	41. language	68. support
15. data	42. learn	69. systems
16. dealing	43. linear	70. teach
17. desktop	thread	71. team
18. different	meeting	72. technologies
place	44. matter	73. technology
19. distance	45. meeting	74. tool
20. distance	46. meetings	75. tools
issues	47. memory	76. train
21. distributed	48. money	77. training
meetings	49. multi-media	78. user
22. distributed	50. network	79. users
23. distributed	51. networks	80. video
environment	52. notes	81. virtual
24. education	53. people	82. voice
25. environment	54. performance	83. voice
26. environments		recognition

session. The ranking scale was:

0 - Too vague, Too specific, or otherwise inappropriate

1 - Somewhat relevant

2 - Most relevant

The instructions plus the numbered 83 terms were listed on a single sheet of paper. Subjects called out the number of the term plus a "0," "1," or "2" to the experimenter. Subjects completed this task in ten to fifteen minutes.

6.1.3 Third Stage: Term Association

The list of 83 terms was converted to a shorter list by combining terms which were singular and plural forms of the same term (e.g., "application" and "applications") and removing terms which did not have semantic content or were obviously too general (e.g., "matter" and "issues"). Term phrases were also combined with individual terms (e.g., "culture," "cultural," and "cultural differences" into "culture/cultural differences"). The numbered list of 59 terms in Table III resulted.

Subjects were given a one-page printout of the numbered 59 terms and a three-column table containing 59 rows. They were asked to compare each of the terms in the first column with the remaining terms on the list and write the numbers of those remaining terms which are somewhat associated in the second column of the worksheet and the numbers of the terms which are strongly associated in the third column. Examples of the word list and work sheet are in Appendix A. This stage required 1,711 pairwise comparisons ($59 \times 58/2$) and assumed that terms were symmetrically associated (if *termA* is associated with *termB*, then, from the perspective of *termB*, *termB* is equally associated with *termA*).

Table III -- List of Association Terms

1. ai	31. memory
2. application/ applications	32. money
3. bandwidth	33. multi-media
4. collaboration/ collaborative systems	34. network/networks
5. communication	35. notes
6. cost	36. people
7. culture/ cultural differences	37. performance
8. data	38. project memory
9. desktop	39. projection
10. different place	40. recognition
11. distance/ distance issues	41. repository
12. distributed environment	42. research methodologies
13. distributed meetings	43. reward
14. education	44. seamless
15. environment/ environments	45. social/societal/ society
16. facilitation/ facilitator	46. standard/standards
17. group/groups	47. style
18. group memory	48. support
19. groupware	49. systems
20. hardware	50. teach
21. human	51. team
22. hypertext	52. technologies/ technology
23. individual	53. tool/tools
24. information	54. train/training
25. integration	55. user/users
26. interpersonal	56. video
27. language	57. virtual
28. learn	58. voice
29. linear thread meeting	59. voice recognition
30. meeting/meetings	

Times to complete this third stage ranged from just under one hour to just under two hours. Data which resulted from this stage included a set of associated terms identified by each of the subjects as appropriate for the EBS question.

6.2 SUBJECTS OF THE EXPERIMENT

Eight subjects completed the three stages. Three of the subjects were expert facilitators. They each had experience facilitating more than 25 meetings over the past two years. Two of the facilitators were "novices" in that they had been trained in facilitation and had each facilitated five to ten meetings. The three remaining subjects had facilitated or participated in conducting many electronic meetings over a two-year period. While not very experienced in guiding groups through the convergent process they were very much aware as observers and participants of the objectives of convergence as well as the difficulties in the process.

6.3 EXPERIMENTAL RESULTS

Results for Stage One of the experiment showed that, as a list of terms, all but two of the subjects ranked the Kohonen list as last. In overall list ranking, each subject selected the facilitator list as the best list. Six subjects rated the Hopfield list second-best. Thus, only two subjects felt that the Kohonen list was better than the Hopfield. Most of the subjects felt that the Kohonen list was more like a list of terms than a list of concepts.

Analysis of the second part of Stage One was performed following standard information retrieval recall and precision techniques. These techniques as well as the analysis techniques used for Stages Two and Three are described more fully below.

Subjects again rated the Facilitator list best at both recall and precision. The subjects rated the Kohonen list better at recall than the Hopfield but poorer in precision. Statistical analysis of the results showed no significant difference among the three lists with respect to Recall but, in Precision, the Kohonen list was significantly worse than the Facilitator list.

Stage 2, Term Relevance, was analyzed by computing the average relevance score given per term for each list by each subject. The three relevance averages for each of the eight subjects was statistically analyzed and the Facilitator list was judged by the subjects to have significantly more relevant terms than both the Hopfield and the Kohonen lists.

To measure association of terms to form concepts, each grouping of terms on each list was used to make a new list of associated term pairs. For example, the facilitator list contains the pairs: video/projection, network/bandwidth, multimedia/hypertext, etc. For the Kohonen map, the meaningful terms of each adjacent region on the map were used to form term pairs. Subject evaluations for the pairings were then summed and the sums were statistically analyzed. The favorable result was that the Kohonen map associated terms as well as the human Facilitator as judged by the eight subjects. The Hopfield score for association of terms was significantly worse than both the Facilitator and Kohonen list.

Details of these analyses are described below.

6.3.1 Kohonen is Less Precise Than Facilitator

For stage one, the relative quality of the three lists was measured by recall and precision methods commonly used in information retrieval. The Kohonen algorithm was found to be less precise than the facilitator's list, however the difference in recall is insignificant. Table IV contains a summary of the data for stage one.

The "Identified" row pertains to the number of concepts identified by the human facilitator, Hopfield Algorithm, and Kohonen SOM (i.e., the number of list items). The "Target" row contains the number of concepts that were contained in each list after the subject added missing concepts and deleted inappropriate concepts. The "Relevant" row contains the number of concepts in the original lists that remained in the corrected lists. "Recall" is the number of relevant concepts divided by the number of target concepts identified by each subject. "Precision" is the number of relevant concepts divided by the number of concepts originally identified in the "Identified" row.

With respect to overall quality, subjects clearly preferred the Facilitator list followed by the Hopfield list followed by the Kohonen. Only two subjects preferred the Kohonen over the Hopfield. The rationale expressed by one of these subjects was

Table IV -- Stage One Results

Subjects	Identified	Facilitator 20	Hopfield 20	Kohonen 27
1	Target Relevant Recall Precision	15 14 93% 70%	20 18 90% 90%	11 9 82% 33%
2	Target Relevant Recall Precision	32 14 44% 70%	32 10 31% 50%	32 11 34% 41%
3	Target Relevant Recall Precision	10 10 100% 50%	10 6 60% 30%	10 7 70% 26%
4	Target Relevant Recall Precision	22 20 91% 100%	22 14 64% 70%	20 18 90% 67%
5	Target Relevant Recall Precision	18 18 100% 90%	19 17 89% 85%	22 21 95% 78%
6	Target Relevant Recall Precision	17 16 94% 80%	16 13 81% 65%	14 13 93% 48%
7	Target Relevant Recall Precision	19 18 95% 90%	15 14 93% 70%	24 24 100% 89%
8	Target Relevant Recall Precision	22 20 91% 100%	17 14 82% 70%	19 16 84% 59%
Average Recall Average Precision		89% 81%	74% 66%	81% 55%

that the 27 concepts in the Kohonen list was not unmanageable and it is preferable to

have more items with which to work. A quote from the protocol of one of the subjects included "It is easier to merge and delete rather than search for missing items."

This rationale is captured in comparing Hopfield and Kohonen values for the average recall and average precision rows of the table. The Kohonen recalled more of the concepts as judged by the subjects but, because of the longer list, precision was worse than the Hopfield. As evidenced by the .89 average recall and .81 average precision, subjects rated the human facilitator's performance better than either of the algorithms' performances.

This result is further supported by statistical analysis of the data. Table V contains the results of statistical analysis of the comparison of the Recall capabilities of the three lists as judged by the subjects. Analysis of variance among the recall scores of the eight subjects indicated that there were no difference among the three lists ($p = .4$) and t-tests between each of the three lists also indicated no significant differences.

However, analysis of variance and t-tests among the precision scores (Table VI) indicated a significant difference between the precision of the Kohonen list and the Facilitator and Hopfield lists.

Table V -- RECALL Statistical Analysis

ANALYSIS OF VARIANCE					
SOURCE	DF	SS	MS	F	p
FACTOR	2	0.0870	0.0435	1.07	0.362
ERROR	21	0.8572	0.0408		
TOTAL	23	0.9442			

INDIVIDUAL 95 PCT CI'S FOR MEAN BASED ON POOLED STDEV			
LEVEL	N	MEAN	STDEV
Facil	8	0.8850	0.1832
Hopfield	8	0.7375	0.2104
Kohonen	8	0.8100	0.2112

POOLED STDEV =	0.2020	0.60	0.75	0.90	1.05
----------------	--------	------	------	------	------

TWOSAMPLE T FOR Facil VS Hopfield
 95 PCT CI FOR MU Facil - MU Hopfield: (-0.06567, 0.3607)
 TTEST MU Facil = MU Hopfield (VS NE): T= 1.50 P=0.16 DF= 13

TWOSAMPLE T FOR Facil VS Kohonen
 95 PCT CI FOR MU Facil - MU Kohonen: (-0.1386, 0.2886)
 TTEST MU Facil = MU Kohonen (VS NE): T= 0.76 P=0.46 DF= 13

TWOSAMPLE T FOR Hopfield VS Kohonen
 95 PCT CI FOR MU Hopfield - MU Kohonen: (-0.3003, 0.1553)
 TTEST MU Hopfield = MU Kohonen (VS NE): T= -0.69 P=0.50 DF= 13

6.3.2 Relevance is a Problem for Both Hopfield and Kohonen

With respect to relevance of the terms used in the lists (Stage 2), the scores given each term by each subject were summed for the terms on each list and an average score per item was computed for each list. Each of the eight subject's average scores for each list was then analyzed statistically using analysis of variance and t-test between each of the lists. The results are listed in Table VII. Overall analysis of variance ($p=.055$) allows for no significant differences at the 95th percentile confidence interval. However, individual t-tests indicate that the terms selected by the Facilitator were significantly better than those selected by either the Hopfield or the Kohonen lists ($p=.042$ and $p=.037$, respectively).

Table VI -- PRECISION Statistical Analysis

ANALYSIS OF VARIANCE					
SOURCE	DF	SS	MS	F	P
FACTOR	2	0.2750	0.1375	3.59	0.046
ERROR	21	0.8040	0.0383		
TOTAL	23	1.0790			

				INDIVIDUAL 95 PCT CI'S FOR MEAN BASED ON POOLED STDEV			
LEVEL	N	MEAN	STDEV	-----+-----+-----+-----+-----			
Facil	8	0.8125	0.1727			(-----*-----)	
Hopfield	8	0.6625	0.1904		(-----*-----)		
Kohonen	8	0.5512	0.2209	(-----*-----)			
POOLED STDEV = 0.1957				0.45	0.60	0.75	0.90

TWOSAMPLE T FOR Facil VS Hopfield
 95 PCT CI FOR MU Facil - MU Hopfield: (-0.04638, 0.3464)
 TTEST MU Facil = MU Hopfield (VS NE): T= 1.65 P=0.12 DF= 13

TWOSAMPLE T FOR Facil VS Kohonen
 95 PCT CI FOR MU Facil - MU Kohonen: (0.04705, 0.4754)
 TTEST MU Facil = MU Kohonen (VS NE): T= 2.64 P=0.021 DF= 13

TWOSAMPLE T FOR Hopfield VS Kohonen
 95 PCT CI FOR MU Hopfield - MU Kohonen: (-0.1115, 0.3340)
 TTEST MU Hopfield = MU Kohonen (VS NE): T= 1.08 P=0.30 DF= 13

Results of this stage confirm the lower scores for Precision that resulted from analysis of Stage One data.

6.3.3 Kohonen Associates Terms as well as the Facilitator

To evaluate Stage Three data we created "lists of term-pairs" identified in the three lists. If a list item contained one or more '/' separators the two terms were placed on this new list of term pairs. A sample of these news lists is given in Table VIII.

If a subject rated a given pair as "somewhat associated" or "strongly associated" a 1 or 2, respectively, was added to the list total for that subject. Total scores of term association were thus created for each list for each subject. These eight groups of

Table VII -- Relevance Statistical Analysis

ANALYSIS OF VARIANCE					
SOURCE	DF	SS	MS	F	p
FACTOR	2	0.723	0.362	3.34	0.055
ERROR	21	2.271	0.108		
TOTAL	23	2.994			

				INDIVIDUAL 95 PCT CI'S FOR MEAN BASED ON POOLED STDEV			
LEVEL	N	MEAN	STDEV	-----+-----+-----+-----+-----			
Facil	8	1.3333	0.2738	(-----*-----)			
Hopfield	8	1.0040	0.3077	(*-----)			
Kohonen	8	0.9357	0.3935	(*-----)			
POOLED STDEV = 0.3289				0.75	1.00	1.25	1.50

TWOSAMPLE T FOR Facil VS Hopfield
 95 PCT CI FOR MU Facil - MU Hopfield: (0.01464, 0.6440)
 TTEST MU Facil = MU Hopfield (VS NE): T= 2.26 P=0.042 DF= 13

TWOSAMPLE T FOR Facil VS Kohonen
 95 PCT CI FOR MU Facil - MU Kohonen: (0.02828, 0.7670)
 TTEST MU Facil = MU Kohonen (VS NE): T= 2.35 P=0.037 DF= 12

TWOSAMPLE T FOR Hopfield VS Kohonen
 95 PCT CI FOR MU Hopfield - MU Kohonen: (-0.3133, 0.4499)
 TTEST MU Hopfield = MU Kohonen (VS NE): T= 0.39 P=0.71 DF= 13

three scores were then subjected to statistical analysis to determine if statistical differences in the scores existed.

Our hypothesis was that the Kohonen self-organizing map would prove superior to the Hopfield list and equal to the human facilitator in associating the terms by their placement on the grid. The statistical analysis report of the Term Association stage (Table IX) clearly demonstrates the significant difference between the Hopfield list and both the Facilitator and Kohonen lists. This supports our hypothesis that the Kohonen would associate terms better and further supports the associative quality of the Kohonen in that no statistical difference was found in the evaluations of associated terms between the Kohonen and Facilitator lists. Thus, the Kohonen approximates the

Table VIII -- Associated Pairs for Each List

Facilitator	Hopfield	Kohonen
video / projection	technology / dealing	technology / culture
network / bandwidth	technology / cultural	technology / tools
multimedia / hypertext	differences	technology / support
group memory / project	dealing / cultural differences	technology / recognition
memory	meetings / distributed	culture / recognition
group memory / repository	meetings / human	culture / video
project memory / repository	distributed / human	networks / video
culture / style	ai / data	hardware / networks
distributed / distance issues /	ai / amounts	hardware / users
distance / different place	data / amounts	hardware / group
cost / money	environments / virtual	hardware / reward
education / train	tools / culture	hardware / tools
education / learn	information / ability	hardware / application
education / teach	video / desktop	hardware / integration
train / learn		hardware / groupware
train / teach		application / tools
learn / teach		application / collaboration
human / people		application / bandwidth
human / user		integration / tools
human / individual		integration / collaboration
human / interpersonal		integration / bandwidth
people / user		groupware / tools
people / individual		groupware / collaboration
people / interpersonal		tools / support
people / user		tools / bandwidth
user / individual		support / facilitator
user / interpersonal		facilitator / collaboration
individual / interpersonal		...

human performance in this respect.

Another interesting and confirming view of this analysis is to treat each subject's output of Stage Three as a domain of associated term pairs. That is, from the potential of 1,711 term pairs, the subjects selected a subset of these pairs which are somewhat or strongly associated with respect to the EBS question. These subsets were then

Table IX -- Term Association Statistical Analysis

```

ANALYSIS OF VARIANCE
SOURCE      DF          SS          MS          F          P
FACTOR      2          5669        2835        26.36        0.000
ERROR      21          2258        108
TOTAL      23          7927

INDIVIDUAL 95 PCT CI'S FOR MEAN
BASED ON POOLED STDEV
-----+-----+-----+-----+-----+
LEVEL      N          MEAN        STDEV
Facil      8          37.00        8.50
Hopfield   8          3.00         2.14
Kohonen    8          34.00       15.68
          (-----*-----)
          (-----*-----)
-----+-----+-----+-----+
POOLED STDEV =      10.37
                      0          15          30          45

TWO SAMPLE T FOR Facil VS Hopfield
95 PCT CI FOR MU Facil - MU Hopfield: ( 26.67, 41.33)
TTEST MU Facil = MU Hopfield (VS NE): T= 10.97 P=0.0000 DF= 7

TWO SAMPLE T FOR Facil VS Kohonen
95 PCT CI FOR MU Facil - MU Kohonen: ( -11.05, 17.05)
TTEST MU Facil = MU Kohonen (VS NE): T= 0.48 P=0.64 DF= 10

TWO SAMPLE T FOR Hopfield VS Kohonen
95 PCT CI FOR MU Hopfield - MU Kohonen: ( -44.23, -17.77)
TTEST MU Hopfield = MU Kohonen (VS NE): T= -5.54 P=0.0009 DF= 7

```

treated as a "Target" lists of associated pairs. The number of "Identified" pairs result from counting the term pairs in Table VIII. then consist of those pairs found in each list: the facilitator list contains 27 pairs; the Hopfield list contains 8; and the Kohonen list contains 71. "Relevant" pairs are those pairs identified by a respective source (Facilitator, Hopfield, or Kohonen) that are also on the subject's target list. Recall and Precision can then be computed as before (recall is the number of relevant pairs divided by the number of target pairs and precision is the number of relevant pairs divided by the number of identified pairs).

Table X shows the results of a recall and precision analysis of these lists of paired terms. The Kohonen pairs equal the expert facilitator's pairs of terms with respect to

Table X -- Association of Identified Pairs of Terms

Subjects	Identified	Facilitator 27	Hopfield 8	Kohonen 71
1	Target Relevant Recall Precision	52 20 38% 74%	52 2 4% 25%	52 30 58% 42%
2	Target Relevant Recall Precision	39 18 46% 67%	39 1 3% 13%	39 20 51% 28%
3	Target Relevant Recall Precision	30 16 53% 60%	30 1 3% 13%	30 13 43% 18%
4	Target Relevant Recall Precision	49 26 53% 96%	49 2 4% 25%	49 21 43% 30%
5	Target Relevant Recall Precision	49 23 47% 85%	49 2 4% 25%	49 24 49% 34%
6	Target Relevant Recall Precision	65 27 42% 100%	65 5 8% 63%	65 33 51% 46%
7	Target Relevant Recall Precision	59 24 41% 89%	59 1 2% 13%	59 34 58% 48%
8	Target Relevant Recall Precision	25 16 64% 59%	25 1 4% 25%	25 8 32% 11%
Average Recall Average Precision		48% 79%	4% 25%	48% 32%

Recall (Average Recall is 48%). The low number of Hopfield pairs seems

significantly worse (4%). The Hopfield fares better in Average Precision (25%) but is less accurate than the Kohonen (32%). However, the facilitator is much superior to both sets of pairs identified by Hopfield and Kohonen in precision.

A statistical analysis of the percentage difference demonstrates that the facilitator is significantly better than the Hopfield and Kohonen algorithms in selecting precise pairs of associated terms but that the Kohonen is not significantly different than the facilitator in recalling potential pairs of associated terms. Table XI contains both the Recall and Precision Minitab output from Analysis of Variance testing and t-tests of the data from Table VIII. T-tests between each of the data sets demonstrate the clear significance between the Hopfield scores and both the Facilitator and Kohonen scores ($p=0.0000$) while there is no significant difference in Recall between the Facilitator and the Kohonen data ($p=0.93$).

The Precision scores in Table VIII similarly demonstrate the clear difference in Precision between the Facilitator score and scores for both the Hopfield and Kohonen paired lists.

The Average Recall score of 48% indicates subjects felt that the facilitator and Kohonen map identified less than half of the potentially associated terms. It remains to be tested whether the complements to the identified term pairs would result in new

Table XI -- Recall/Precision Statistical Analysis of Associated Pairs**RECALL:****ANALYSIS OF VARIANCE**

SOURCE	DF	SS	MS	F	p
FACTOR	2	1.02966	0.51483	110.05	0.000
ERROR	21	0.09824	0.00468		
TOTAL	23	1.12790			

**INDIVIDUAL 95 PCT CI'S FOR MEAN
BASED ON POOLED STDEV**

LEVEL	N	MEAN	STDEV	
Facil	8	0.47750	0.07888	(---*---)
Hopfield	8	0.04000	0.01773	(---*---)
Kohonen	8	0.48125	0.08659	(---*---)

POOLED STDEV = 0.06840 0.00 0.15 0.30 0.45

TWOSAMPLE T FOR Facil VS Hopfield

95 PCT CI FOR MU Facil - MU Hopfield: (0.3699, 0.5051)

TTEST MU Facil = MU Hopfield (VS NE): T= 15.31 P=0.0000 DF= 7

TWOSAMPLE T FOR Facil VS Kohonen

95 PCT CI FOR MU Facil - MU Kohonen: (-0.09324, 0.08574)

TTEST MU Facil = MU Kohonen (VS NE): T= -0.09 P=0.93 DF= 13

TWOSAMPLE T FOR Hopfield VS Kohonen

95 PCT CI FOR MU Hopfield - MU Kohonen: (-0.5152, -0.3673)

TTEST MU Hopfield = MU Kohonen (VS NE): T= -14.12 P=0.0000 DF= 7

PRECISION:**ANALYSIS OF VARIANCE**

SOURCE	DF	SS	MS	F	p
FACTOR	2	1.3556	0.6778	29.14	0.000
ERROR	21	0.4884	0.0233		
TOTAL	23	1.8440			

**INDIVIDUAL 95 PCT CI'S FOR MEAN
BASED ON POOLED STDEV**

LEVEL	N	MEAN	STDEV	
Facil	8	0.7875	0.1602	(---*---)
Hopfield	8	0.2525	0.1637	(---*---)
Kohonen	8	0.3212	0.1316	(---*---)

POOLED STDEV = 0.1525 0.25 0.50 0.75 1.00

TWOSAMPLE T FOR Facil VS Hopfield

95 PCT CI FOR MU Facil - MU Hopfield: (0.3600, 0.7100)

TTEST MU Facil = MU Hopfield (VS NE): T= 6.61 P=0.0000 DF= 13

TWOSAMPLE T FOR Facil VS Kohonen

95 PCT CI FOR MU Facil - MU Kohonen: (0.3079, 0.6246)

TTEST MU Facil = MU Kohonen (VS NE): T= 6.36 P=0.0000 DF= 13

TWOSAMPLE T FOR Hopfield VS Kohonen

95 PCT CI FOR MU Hopfield - MU Kohonen: (-0.2292, 0.09173)

TTEST MU Hopfield = MU Kohonen (VS NE): T= -0.93 P=0.37 DF= 13

concepts for the topic list or not.

CHAPTER 7: CONCLUSIONS, CONTRIBUTIONS, AND FUTURE DIRECTION

We can conclude from this research that significant progress can be made in supporting a complex human problem-solving task with software. In particular the Kohonen Self-Organizing Map can be used to Recall terms to be used for classification of Electronic Brainstorming comments at human performance levels. Further conclusions are discussed below. This research contributes to the solution of managing the complexities involved with noisy, textual data by suggesting a graphical representation of the existing data which is generalizable in that the data itself determines the representation. Future research directions include actually generalizing the application to other electronic meeting applications as well as applications where large sets of textual data are found.

7.1 CONCLUSIONS

This research demonstrates how software can be used to perform an intelligent classification of Electronic Brainstorming (EBS) output. The measure of performance that we use is comparing the software's performance against a human expert facilitator's performance in solving the same complex problem as well as previous work using a Hopfield neural network algorithm. Specific conclusions from this research include: Recall of concepts by the Kohonen is no different than that of the human expert nor the output of the Hopfield algorithm. Those concepts produced by

the Kohonen algorithm are less precise than those produced by the human facilitator but not any different from those produced by the Hopfield algorithm. With respect to relevance of the terms in connection with the electronic brainstorming question itself both the Kohonen and Hopfield algorithms suffer with respect to the facilitator's selection of terms. Finally, with respect to association of terms the Kohonen algorithm performs as well as the expert facilitator.

Starting with a suggested list of concepts for classification of electronic brainstorming comments, even if somewhat imprecise, is much better than starting with a blank screen and hundreds of lines of comments. This research demonstrated that recall of concepts is no different than those concepts offered by a human expert facilitator or the Hopfield neural network algorithm.

However, the concepts offered by an expert facilitator were more precise (a higher proportion of relevance) than those concepts offered by the Kohonen and Hopfield algorithms. No difference exists between the precision of the concept lists produced between the Kohonen and Hopfield algorithms.

With regard to the individual terms which make up the concept clusters or individual concept list items both Kohonen and Hopfield algorithms tend to suggest

terms which are not very relevant to the electronic brainstorming topic. Again, recall of terms, even if somewhat irrelevant, is better than starting with nothing at all.

Where the Kohonen shows the most promise is in the association of terms that may suggest formation of concept clusters. This research found that the Kohonen algorithm associated terms as well as a human facilitator and much better than the Hopfield algorithm. The graphical output and juxtaposition of terms on this graphical representation suggest associations and a spectrum of potential topics.

7.2 CONTRIBUTIONS

The principle contribution of this research is the addition of a graphical approach to EBS output data management. This graphical approach greatly reduces the complexity of synthesizing large quantities of EBS output into manageable subsets that are classified according to their similarity of content. This new dimension of data management provides a useful contribution to the group problem solving process in suggesting a candidate classification scheme that has been applied in a consistent manner. Groups will find it much easier to adjust a proposed classification scheme than it is to start with an empty list.

A further contribution is that the approach should be generalizable to not only other Electronic Meeting tool output but to other noisy, text-based applications.

Electronic mail collections, internet usenet discussions, bibliographic data bases, and digital library collections are possible domains for application of the Kohonen self-organizing map.

7.3 FUTURE DIRECTIONS

The immediate future direction of this research needs to continue focusing on how comments are clustered for classification purposes. This work has focused on associating terms which are candidates for topic headers due to their relevance and degree of association with respect to the subject area. However, it has not focused on the actual classification of the comments themselves. One of the major differences between the Hopfield and Kohonen algorithms is that comments are attached to Hopfield topics by matching the keywords contained within those topics. Comments containing more than one keyword identified in the Hopfield topic list will be attached to each topic that matches the keywords. Thus, one comment might appear within more than one topic heading. The Kohonen algorithm places each comment in its "best" location. More research is needed to determine which approach is more appropriate in the electronic brainstorming classification problem domain.

More work is also needed in enhancing the user interface. Better real-time classification support should be provided in order to allow adjustment of the dimensions of the Kohonen map. Users will need to be able to browse the underlying

comments/text within map regions to apply better labels to the regions, merge areas of the map into one conceptual region, or place comments/text in regions that they may feel are more appropriate. Integration of the Kohonen algorithm into the categorization electronic meeting tool would be a first step.

Electronic brainstorming is only one of several divergent tools in the electronic meeting system tool set. The automatic indexing portion of the algorithm must be expanded to allow input from other electronic meeting tool formats such as a topic commenter or group outliner. This will also add more flexibility to using this tool over time as an intelligent meeting memory system as meeting data is collected over the life of a project within organizations.

Further generalization to organizational memory applications is needed. Integration executive information system applications may aid in organizational learning and better strategic planning by providing a graphical view of the relationships of organizational textual data.

The graphical representation of the Kohonen SOM provides an abstraction that should be useful for large information spaces such as digital libraries, internet usenet discussions, or other large textual data collections. These larger collections may

require extending the Kohonen algorithm to provide hierarchical layers of maps to aid in information retrieval and/or browsing of these larger collections.

APPENDICES

APPENDIX A -- EXPERIMENT HANDOUT

The following 11 pages were given to each of eight subjects. The purpose of the experiment was to evaluate the concept lists generated by an expert human facilitator, the Hopfield algorithm, and the Kohonen Self-Organizing Map (SOM). The evaluation consisted of three stages: Concept list comparisons (whole list), relevance of terms used on the lists, and degree of associativity among terms on the list.

Concept Categorization Experiment April 1994

Purpose of the experiment: Given a sample of brainstorming output, objectively evaluate the quality of lists of concepts by:

1. evaluating the quality of three sample lists,
2. determining degrees of relevance of representative keywords,
3. and determining degrees of association between the keywords.

Thus, the experiment consists of three major parts:

- A. List evaluation
- B. Relevance evaluation
- C. and Keyword association.

In List Evaluation you will read through a set of brainstorming output, rank order three representative lists of concepts that were generated from that output, and correct the three lists.

In Relevance evaluation you will be given a list of keywords from the electronic brainstorming output and be asked to evaluate the relevance of each of the terms. (0=irrelevant, 1=somewhat relevant, 2=most relevant)

For Keyword association you will compare each of the keywords with the remaining keywords to evaluate how strongly they are related. (0=unrelated, 1=somewhat related, 2=strongly related)

Concept Categorization Experiment
April 1994

Part A -- List Evaluation

1. Put the following lists in a rank order and
2. Alter each list (adding/deleting concepts) to make them better.

Setting:

1. Assume that these lists were created by different groups of people from their analysis of electronic brainstorming (EBS) comments. The EBS question was:

"What are the most important technology problems with respect to Collaborative Systems to be solved over the next five years?"

2. The goal of the categorization process was to derive the lists of concepts for use in rank order vote. The voting issue will be "In what order should we address the problem areas?"

3. Assume that you are a facilitator and will lead a group through the voting.

Directions:

1. State the criteria that you will use to evaluate the lists. As a facilitator, what makes a good list? (Response will be tape-recorded.)
2. Browse through the printout of the EBS session. Feel free to take notes if you wish.
3. Rank the lists from 1 to 3 with 1 being the best and 3 being the least appropriate list.
4. "Correct" each list:
 - A. Treat each line item on each list as a concept.
 - B. Delete inappropriate lines, add missing appropriate concepts.
 - C. If you do not like the wording of the concept use a delete and add procedure.
5. Review the list ranking.
6. Review the evaluation criteria that you used. (Response will be tape-recorded)

[illegible]

Facilitator: _____

Rank _____

Date/Time: _____

technology

tools

meeting

support

collaborative

facilitator

matter

systems

people / issues / communication

application / integration / groupware

recognition

cultural

video

hardware

networks

dealing

users / group / reward

collaboration / seamless

memory / tool

training

virtual

environment

notes

information / ability

distributed

standard

bandwidth

**Concept Categorization Experiment
April 1994**

Part B -- Relevance Evaluation

The following list contains keywords extracted from the electronic brainstorming output (hard copy is attached). Please rate the importance of these terms with respect to whether they should be included within a concept list pertaining to the electronic brainstorming session. The ranking scale should be:

- 0 - Too vague, Too specific, or otherwise inappropriate
- 1 - Somewhat relevant
- 2 - Most relevant

1. ability	33. hardware	66. standards
2. ai	34. human	67. style
3. amounts	35. hypertext	68. support
4. application	36. individual	69. systems
5. applications	37. information	70. teach
6. bandwidth	38. integration	71. team
7. collaboration	39. interpersonal	72. technologies
8. collaborative	40. issues	73. technology
9. collaborative
systems	41. language	74. tool
10. communication	42. learn	75. tools
11. cost	43. linear thread
meeting	76. train
12. cultural	44. matter	77. training
13. cultural differences	45. meeting	78. user
14. culture	46. meetings	79. users
15. data	47. memory	80. video
16. dealing	48. money	81. virtual
17. desktop	49. multi-media	82. voice
18. different place	50. network	83. voice recognition
19. distance	51. networks	
20. distance issues	52. notes	
21. distributed
meetings	53. people	
22. distributed	54. performance	
23. distributed
environment	55. project memory	
24. education	56. projection	
25. environment	57. recognition	
26. environments	58. repository	
27. facilitation	59. research
methodologies	
28. facilitator	60. reward	
29. group memory	61. seamless	
30. group	62. social	
31. groups	63. societal	
32. groupware	64. society	
	65. standard	

**Concept Categorization Experiment
April 1994**

Part C -- Keyword Association

The purpose of this experiment is to determine the degree of association among the keywords extracted from the electronic brainstorming comments. Please identify the terms that are somewhat or strongly related to each of the following terms. Write in the number of the term using the **Terms List** attached to the following table.

For example, if you feel "ai" is somewhat related to "applications" enter "2" in the **Somewhat Related** box. If it is strongly related, enter the "2" into the **Strongly Related** box.

"Relatedness" means with respect to creating clusters of terms to represent concepts represented in the electronic brainstorming question and comments.

Term	Somewhat Related	Strongly Related
ai		
application/ applications		
bandwidth		
collaboration/ collaborative systems		
communication		
cost		
culture/cultural differences		
data		
desktop		
different place		

distance/distance issues		
distributed environment		
distributed meetings		
education		
environment/ environments		
facilitation/ facilitator		
group/groups		
group memory		
groupware		
hardware		
human		
hypertext		
individual		
information		
integration		
interpersonal		
language		

learn		
linear thread meeting		
meeting/meetings		
memory		
money		
multi-media		
network/ networks		
notes		
people		
performance		
project memory		
projection		
recognition		
repository		
research methodologies		
reward		
seamless		

social/societal/ society		
standard/ standards		
style		
support		
systems		
teach		
team		
technologies/ technology		
tool/tools		
train/training		
user/users		
video		
virtual		
voice		
voice recognition		

Terms List

1. ai
2. application/applications
3. bandwidth
4. collaboration/collaborative systems
5. communication
6. cost
7. culture/cultural differences
8. data
9. desktop
10. different place
11. distance/distance issues
12. distributed environment
13. distributed meetings
14. education
15. environment/environments
16. facilitation/facilitator
17. group/groups
18. group memory
19. groupware
20. hardware
21. human
22. hypertext
23. individual
24. information
25. integration
26. interpersonal
27. language
28. learn
29. linear thread meeting
30. meeting/meetings
31. memory
32. money
33. multi-media
34. network/networks
35. notes
36. people
37. performance
38. project memory
39. projection
40. recognition
41. repository
42. research methodologies
43. reward
44. seamless
45. social/societal/society
46. standard/standards
47. style
48. support
49. systems
50. teach
51. team
52. technologies/technology
53. tool/tools
54. train/training
55. user/users
56. video
57. virtual
58. voice
59. voice recognition

APPENDIX B -- PASCAL CODE: COMMENT CONVERSION TO KOHONEN INPUT FORMAT

```

program ebs2kohonen( input, output );

type
  lineStr   =      string[80];
  termStr   =      string[30];
  termPtr   =      ^termRcd;
  termRcd   =      record
                        termPhrase   :   termStr;
                        termCount    :   integer;
                        next         :   termPtr;
                      end;

var
  infile, infile2, outfile      :   text;
  inFileNam, inFileNam2, outFileNam :   lineStr;
  threshold, code               :   integer; { doc freq lvl }
  lineIn      :   lineStr; { use for input }
  term        :   termStr; { keyword(s)/phrase }
  termTable   :   termPtr; { ptr to table }
  endPtr      :   termPtr; { table endnode }
  count, freq :   integer; { term ctr/term freq }
  numTerms   :   integer; { total num of terms }
  docNum     :   integer; { index comment num }
  docNumHold :   integer; { hold comment num }
  firstTime  :   boolean; { iteration flag }

procedure displayTable ( termTable : termPtr );
var
  myPtr : termPtr;
begin
  { check to see if table is okay: }
  writeln ( 'Display Table: ==>' );
  writeln ( '      Press ENTER' );
  readln (input);
  if termTable = NIL then writeln ( 'NIL table!' )
  else begin
    count := 0;
    myPtr := termTable;
    while myPtr <> NIL do
      begin
        count := count + 1;
        writeln ( count, ': ', myPtr^.termPhrase, ': ', myPtr^.termCount );
        myPtr := myPtr^.next;
      end; { while myPtr <> NIL }
    end; { else }
  end; { displayTable procedure }

procedure parseFreqLine ( lineIn : lineStr;
                        var term : termStr;
                        var count, freq : integer);
var
  cstr, fstr :      string[10]; { count/freq input strings }
  rfreq      :      real;
  ptr, i     :      integer;

begin
  cstr := '';
  fstr := '';

  freq := MAXINT; { for input file without freq information }

```

```

cstr := copy ( lineIn, 1, 3 );
term := copy ( lineIn, 5, length(lineIn));
Val(cstr, count, i);           { convert string to int }
if i <> 0 then
begin
  Writeln('Error at position: ', i);
  readln (input);
end;

(* This commented code is from previous file format:
   eg.:      term[space][space]df[0]=124.00000

ptr := pos ( ' ', lineIn );      { where's the end of term? }
term := copy ( lineIn, 1, (ptr-1) );
ptr := pos ( 'df[', lineIn );
i := ptr + 3;
ptr := pos ( ']', lineIn );      { ptr points to ']' }
cstr := copy ( lineIn, i, (ptr - i) );
Val(cstr, count, i);           { convert string to int }
if i <> 0 then
begin
  Writeln('Error at position: ', i);
  readln (input);
end;
fstr := copy ( lineIn, (ptr + 2), length(lineIn));
{ ptr still at ']' }
Val(fstr, rfreq, i);
if i <> 0 then
begin
  writeln ('frequency conv. error at: ', i);
  readln (input);
end;
freq := trunc(rfreq);

*)

end; { parseFreqLine procedure }

procedure initTable ( var termTable : termPtr );
var
  myPtr : termPtr;

begin
  myPtr := termTable;
  while myPtr <> NIL do
  begin
    myPtr^.termCount := 0;
    myPtr := myPtr^.next;
  end; { while endPtr <> NIL }
end;

procedure parseIndexLine ( lineIn : lineStr;
                           var term : termStr;
                           var docNum : integer);
var
  docstr      : string[10]; { doc input string }
  ptr, i, j   : integer;

begin
  docstr := '';
  ptr := pos ( '.', lineIn );      { where's the decimal? }
  i := ptr + 1;                    { point to the number }
  ptr := pos ( ' ', lineIn );
  docstr := copy ( lineIn, i, (ptr-i)); { grab the number }

```

```

{ move the pointer to the term.  jump over 4 numbers: }
{ paragraph #, sentence #, word #, number of words in phrase }
{ currently ptr points to first space before first number }
for j := 1 to 5 do
begin
  lineIn := copy ( lineIn, ptr+1, length(lineIn) );
  ptr := pos ( ' ', lineIn );
end; { for j := 1 to 4 }
term := lineIn;
Val(docstr, docNum, i);           { convert string to int }
if i <> 0 then
begin
  Writeln('Error at position: ', i);
  readln (input);
end;
end; {parseIndexLine Procedure }

procedure addToTable ( var termTable, endPtr : termPtr;
                      term : termStr );
var
  tempPtr : termPtr;

begin
  new(tempPtr);
  with tempPtr^ do
  begin
    termPhrase := term;
    termCount := 0;
    next := NIL;
  end; { with tempPtr }

  if termTable = NIL then { check for first time }
  begin
    termTable := tempPtr;
    endPtr := tempPtr;
  end
  else begin
    endPtr^.next := tempPtr;
    endPtr := tempPtr;
  end; { else termTable <> NIL }
end; { addToTable Procedure }

procedure updateTable (var termTable : termPtr;
                      term : termStr);
var
  tempPtr : termPtr;

begin
  tempPtr := termTable;

  while tempPtr <> NIL do
  begin
    if (term = tempPtr^.termPhrase) then
    begin
      { Later Research: test whether to just make value 1 }
      { or add number of occurrences }
      tempPtr^.termCount := 1;
      tempPtr^.termCount := tempPtr^.termCount + 1; }
      end; { if terms are equal }
      tempPtr := tempPtr^.next;
    end; { while }
  end; { updateTable procedure }

  procedure writeOut ( termTable : termPtr );

```



```

var
  myPtr : termPtr;

begin
  myPtr := termTable;
  while myPtr <> NIL do
    begin
      write (outFile, myPtr^.termCount, '.0 ');
      myPtr := myPtr^.next;
    end; { while }
    writeln (outFile);
  end; { writeOut procedure }

  (*=====*)
begin { main -- read in the freq list table/process the index file }

(* paramStr's: 1=freq list file,
               2=index filename,
               3=outputfile,
               4=threshold
*)
  if ParamCount < 4 then
    begin
      Writeln('Incorrect # of parameters supplied');
      writeln('PGMName inputfileName outputfileName threshold');
      exit
    end;
  inFileNam := ParamStr(1);           { freq list with thresholds }
  inFileNam2 := ParamStr(2);          { index file }
  outFileNam := ParamStr(3);          { output file }
  Val(ParamStr(4), threshold, code); { threshold value }
  if code <> 0 then
    begin
      Writeln('Threshold Conversion Error at: ', code);
      exit
    end;

  assign (infile, inFileNam);
  reset (infile);
  termTable := NIL;
  endPtr := NIL;
  term := '';
  freq := MAXINT;
  writeln; writeln ( '==>' );
  writeln ( 'Threshold: ', threshold);

{ main loop }
  { build the frequency list table according to threshold set }
  while ((not EOF(infile)){ and (freq >= threshold)}) do
    begin
      readln(infile, lineIn);
      { go get the term string, seq. term count, and doc freq. }
      parseFreqLine(lineIn, term, count, freq);
      if (freq >= threshold) then
        begin
          addToTable ( termTable, endPtr, term );
        end;
    end; { while not EOF of frequency list }
  numTerms := count;

{ read in the index file and write out a line at a time to output }
  close (infile);
  assign (infile, inFileNam2);

```

```

reset (infile);
assign (outfile, outFileNam);
rewrite (outfile);
writeln (outfile, numTerms);
docNumHold := 0;
firstTime := true;

while (not EOF(infile)) do
begin
  readln (infile, lineIn);
  parseIndexLine (lineIn, term, docNum);
  { docNum keys to a new comment when changes }
  { '1.15 to 1.16' or '1.24 to 2.1' - we only look at change }
  { in the decimal part (docNum is the decimal part) }
  if ((docNum <> docNumHold) and (not firstTime)) then
  begin
    { write output }
    writeOut ( termTable );

    { initialize doc vector to zeroes. }
    initTable (termTable);
    docNumHold := docNum;
    end; { if docNum <> docNum }
    updateTable (termTable, term);
    firstTime := false;
  end; { while not EOF infile }
writeOut ( termTable );
close ( infile );
close ( outfile );
end.

```

APPENDIX C -- PASCAL CODE: CREATE A SET OF UNIT VECTORS

(*

This program creates an input file for the Kohonen SOM. The file created is of this format:

The list of terms for the experiment contained 190 terms and term phrases. Each unit vector represents one of these terms or term phrases. This file of 190 unit vectors was used twice. Once to locate the map coordinate that is closest to each term and another time as a map itself. As a map, the node values of the Kohonen grid are compared to this unit vector map to assign a term or term phrase to each node on the Kohonen grid. This determines the map boundaries between terms and term phrases.

*)

```

program unitvect ( input, output );

var
  i, j      : integer;
  outFile   : text;

begin
  assign ( outFile, 'unit190.vct' );
  rewrite ( outFile );

  writeln ( outFile, '190' );
  for i := 1 to 190 do
    begin
      for j := 1 to 190 do
        if i=j
          then write ( outFile, '1.0 ' )
          else write ( outFile, '0.0 ' );
        writeln ( outFile );
      end;
    end;
  close ( outFile );
end.

```

APPENDIX D -- PROCESS FOR RUNNING KOHONEN SOM ON EBS OUTPUT

Building upon the work of the Hopfield Algorithm, we replaced the Concept Space Generation and Classification stages with Kohonen Self-Organizing Map (SOM) software. The output of the Automatic Indexing stage included the index of terms contained within the comment file plus the list of terms and term phrases in order of decreasing frequency. This index and list of terms was used to create the file of input vectors needed for the Kohonen SOM. A two-dimensional grid of vectors was created and initialized to random values. The input file was repeatedly run against the grid following these general steps: 1) Compare the current input vector with each vector on the grid. 2) Find the grid vector which is closest ("winner") to the input vector. 3) Compute an adjustment factor and adjust the winner to more closely resemble the input vector. 4) Adjust each vector in a defined area (neighborhood) around the winner by a like amount. After thousands of iterations through the input file the adjusted vector values on the two-dimensional grid were used to map each comment onto the one grid vector with which it is most closely related. Thus, all of the comments assigned to the same grid coordinate were related. The list of terms and term phrases was also used with the map to determine the most best term or term phrase to assign to each node on the grid. Assignment of terms and term phrases to grid nodes resulted in identifiable topic regions with seemingly related terms in adjacent regions as shown in Figure 28.

```

1.1 1 1 2 1 DEVELOPMENT
1.1 1 1 4 1 STANDARD
1.1 1 1 5 1 USER
1.1 1 1 5 2 USER INTERFACES
1.1 1 1 6 1 INTERFACES
1.1 1 1 8 1 REOMTE
1.1 1 1 11 1 INFO
1.1 1 1 13 1 COMMUNICATION
1.2 1 1 3 1 XYZ
1.2 1 1 3 2 XYZ NOTES
.
.
.
30.7 1 1 3 1 MATTER
30.7 1 1 10 1 LIMITED
30.7 1 1 17 1 LIMITED
30.7 1 2 19 1 GOD
30.7 1 2 21 1 UNANSWERED
30.7 1 2 21 2 UNANSWERED PRAYERS
30.7 1 2 22 1 PRAYERS

```

Figure 21 -- Indexing output examples

We use the same Automatic Indexing output that is generated in the indexing stage of the Hopfield algorithm. From our example, 212 comments are indexed by the terms and term phrases contained within the comments and 190 terms and term phrases were selected. The index appears as in Figure 21.

Each line of the file indexes a term or term phrase of the set of comments. The first number on each line references the unique comment number assigned when the comment was created in EBS. The second number refers to the paragraph number within the comment where the term can be found. The third number refers to the

number of the sentence in which the term or term phrase is located. The fourth number refers to the number of words from the beginning of the comment to the term or term phrase.

0	TECHNOLOGY
1	MEETINGS
2	COLLABORATIVE SYSTEMS
3	SUPPORT
4	TOOLS
5	COLLABORATIVE
6	SYSTEMS
7	MEETING
8	DISTRIBUTED
9	PEOPLE
10	ENVIRONMENTS
11	TECHNOLOGIES
12	ENVIRONMENT
13	INFORMATION
14	VIDEO
15	NETWORKS
16	ISSUES
17	NETWORK
18	LINEAR THREAD MEETING
19	HARDWARE
20	DISTRIBUTED MEETINGS
21	PROCESSES
22	CULTURAL
...	
181	CHANGE
182	WORLD
183	COMMON SPOKEN LANGUAGE
184	DIFFERENCES
185	REMOTE COLLABRATIVE MEETINGS
186	WINDOWS
187	REMOTE SCREEN SHARING
188	MODE
189	RELIABILITY
190	LIMITED

Figure 22 -- Terms List in Decreasing Order of Frequency

Removal of stop words (words such as "the," "which," and other weak semantic terms) resulted in a list of 190 terms and term phrases in decreasing order of frequency of appearance (Figure 22).

The index and this list of terms and term phrases were used as input to a parsing program which generated the input file for use with the Kohonen Self-Organizing Map (SOM). The Pascal code for this parsing program is listed in Appendix B. The format of the Kohonen input in our sample case was 213 rows of 190 real numbers per row. The 213 rows represent a Kohonen algorithm setup row

190	hexa	10	20	bubble			
0.0	0.0	0.0	0.0	...	0.0	!	Comment 1.1
0.0	0.0	0.0	0.0	...	0.0	!	Comment 1.2
0.0	0.0	0.0	0.0	...	0.0	!	Comment 1.3
...							
0.0	0.0	0.0	0.0	...	0.0	!	Comment 30.7

Figure 23 -- Sample Kohonen Input Data

followed by 212 rows -- each row representing a comment. Figure 23 provides a sample of the input file. The 190 real numbers for each comment row represents a vector of the most meaningful and frequently-used 190 terms and term phrases. The first vector value represents *technology*, the second term represents *meetings*, and so on. The actual value of each vector value is 0.0 if the represented term is not contained within the comment and 1.0 if it is contained within the comment.

The first row contains a list of parameters that the Kohonen software should use (190 hexa 10 20 bubble). "190" indicates that there are 190 terms to be classified.

"Hexa" indicates that the pattern, or lattice-type, of the neighborhood of each node on the grid should be viewed hexagonally instead of rectangularly (six immediate neighbors instead of eight). The difference between a hexagonal and rectangular neighborhood for a 3 x 7 matrix is shown in Figure 24.

Figure 24 -- Neighborhood Arrangements

The rationale for choosing the size of the matrix is based mostly upon practicality of the resulting map display and the desired run-time of the program. The resulting map should be viewable on one computer screen and the desired run-time is within 15 minutes. A larger matrix will require a longer run-time. In the case of this

experiment, we selected the number of rows in the map to be 10 and the number of columns to be 20. Each node on this grid is a vector of 190 elements and is also referred to as a "reference vector."

Finally, two functions are available to be used in determining the adjustments to be made to each node -- a bubble function or a Gaussian function. In this case the program will use the bubble function.

First stage: Map initialization

The reference vectors of the map are first initialized to tentative values. The lattice type of the map (hexa) and the neighborhood (bubble) function used in the training procedures are also defined in the initialization. The map size used was 10 by 20 units.

Second stage: Map training

Training is done in two phases. The first of them is the ordering phase during which the reference vectors of the map units are ordered. During the second phase the values of the reference vectors are fine-tuned.

In the beginning the neighborhood radius is taken almost equal to the diameter of the map and decreases to one during training, while the learning rate decreases to

zero. During the second phase the reference vectors in each unit converge to their 'correct' values. The second phase is longer than the first one (10,000 input comparisons instead of 1,000). The learning rate is thereby smaller. The neighborhood radius is also smaller on the average: in the beginning the units up to a distance of three are covered. After these two phases of training the map is ready to be used testing and in analyzing the data.

Third stage (testing): Evaluation of the quantization error

When the entries in the map have been trained to their final values, the resulting quantization error is evaluated. The training file is used for this purpose. This stage computes the quantization error over all the samples in the data file. The output of this stage can be used to compare with other data sets for reliability purposes.

Fourth stage: Determine a map coordinate for each comment

The original data file is run against the map to determine the map coordinate that is most closely related to the comment by virtue of the difference between the input vector and each reference vector on the grid. The output of this stage of the program is shown in Figure 25.

```

3 hexa 10 20 bubble
9 8 0.0454621
7 13 1.45146
9 13 1.11768
9 13 1.16124
9 13 0.985868
5 1 1.49492
1 3 1.34818
0 19 1.20812
3 6 1.58436
0 2 1.52031
6 6 0.85493
9 8 0.0454621
9 8 0.0454621
7 4 1.10765
9 19 1.91795
...
8 6 1.32672
0 9 0.898638
0 10 1.97179
6 7 0.96835
9 19 1.9195
6 7 0.967335
4 5 1.31115
4 5 0.93199
3 10 0.789412

```

Figure 25 -- Assignment of Comments to Grid

In our example there are 213 lines in the output. The first line format describes the parameters used in the mapping. Each of the following 212 lines represents one comment in the set of comments. The first two columns indicate the location by row and column, respectively, of the node on the Kohonen map which is most closely related to the comment. The third column is the quantization error between the vector of the comment and the reference vector that was selected. Thus, the first comment (1.1) is most closely related to the node located at row 9, column 8,

and the relationship is very strong as reflected in the small quantization error (.0454621).

Fifth stage: Determine a map coordinate for each term

A file of vectors was created that contained a single 1.0 in each row

190	hexa	10	20	bubble			
1.0	0.0	0.0	0.0	...	0.0	!	Term 1
0.0	1.0	0.0	0.0	...	0.0	!	Term 2
0.0	0.0	1.0	0.0	...	0.0	!	Term 3
...							
0.0	0.0	0.0	0.0	...	1.0	!	Term 190

Figure 26 -- Unit Vector Input for Mapping Terms to the Grid

(Figure 26). Each row thus represented one of the 190 terms.

This input was run against the map to determine the nodes most closely related to each of the terms in the terms list. The output file format is identical with that of the Fourth stage.

Sixth stage: Assign a term to each coordinate

In order to find the term or term phrase which is most closely related to each node on the map we reversed the files of the Fifth stage. We treated the file of unit vectors as a (190 x 1)-grid map and used the program to run each reference vector of the map against the unit vector. Thus, this picked which of the 190 terms was most

```

3 hexa 190 1 bubble
0 0 1.20097
0 0 1.11263
0 0 1.10374
22 0 1.09372
11 0 1.02806
142 0 1.02931
142 0 1.05836
14 0 0.999702
14 0 0.689409
14 0 0.71367
0 0 0.834444
0 0 0.701403
22 0 1.03529
11 0 0.89544
11 0 0.790905
142 0 1.02861
22 0 1.04928
...
1 0 0.773404
1 0 0.921331
8 0 0.960753
8 0 0.769595
8 0 0.834672

```

Figure 27 -- Grid Assignment of Terms to the Map

closely related to each node on the map. The resulting output was in the format shown in Figure 27.

The input file to this stage (the original map) contained 201 rows (one header line and 200 vectors of 190 elements). The order of this input file was node (0,0), (1,0), (2,0), ... (9,0), (0,1), (1,1), ... (9,19). Each row of the output file points to the term number on the 190-term list which is most closely related to that node. The first three nodes of output, (0,0), (1,0), and (2,0) are most closely related to "technology,"

the first term on the list. The vector of the fourth node, (3,0), is most closely related to the unit vector for the term "cultural." Labeling each node with its related term results in the Kohonen map represented in Figure 28.

Figure 28 -- Kohonen Self-Organizing Map -- 10 x 20 grid

The Kohonen map is a grid of nodes or pixels with coordinates such that Pixel (0,0) is in the upper left corner and pixel (9,19) is in the lower right corner. Figure 28 was constructed by using the output of Stage Six and labeling each pixel. Once all of

the pixels were labeled, a line was drawn between all of the pixels which had dissimilar terms.

```

MAKEFILE for Kohonen SOM
##
## Self-Organizing Map Program Package
##
##   makefile for UNIX with SysV or GNU make utility
##
##   copy this file to the name makefile before executing the make command
##   if you are using GNU gmake, change the definition of $(STRANGE)
##
## version 1.0 October 1992
##
## NOTE: This program package is copyrighted in the sense that it
## may be used for scientific purposes. The package as a whole, or
## parts thereof, cannot be included or used in any commercial
## application without written permission granted by its producers.
## No programs contained in this package may be copied for commercial
## distribution.
##
## All comments concerning this program package may be sent to the
## e-mail address 'lvq@cochlea.hut.fi'.
##
## platform specific definitions:

SHELL = /bin/csh
CC = cc
CFLAGS = -O
LDFLAGS =
LDLIBS = -lm
GRAPHICSLIBS = -lgl_s

# the following is used with SysV make utility
STRANGE = $$@.c

# the following is used with GNU implementation of make (gmake)
#STRANGE = %: %.c

###

NONGRAPHICTARGETS = vsom randinit lininit visual vcal qerror sammon vfind
GRAPHICTARGETS = gvisual

TARGETS = $(NONGRAPHICTARGETS) $(GRAPHICTARGETS)

ROUTINES = lvq_pak.o som_rout.o
HEADERS = lvq_pak.h

all : $(NONGRAPHICTARGETS) graphics

.c.o : ; $(CC) $(CFLAGS) -c $.c

.c : ; $(CC) $(CFLAGS) $(LDFLAGS) $@.c $(ROUTINES) $(LDLIBS) -o $@

$(ROUTINES) : $(HEADERS) makefile

$(TARGETS) : $(STRANGE) $(ROUTINES)
    $(CC) $(CFLAGS) $(LDFLAGS) $@.c $(ROUTINES) $(LDLIBS) -o $@

graphics:
    @-if ( -x /bin/4d && { /bin/4d } && \
        ! { $(MAKE) -q $(GRAPHICTARGETS) } ) \
        $(MAKE) LDLIBS="$(GRAPHICSLIBS) $(LDLIBS)" $(GRAPHICTARGETS)

```

```

#      Rename your input data file to ex.dat to
#      run this and modify the
#      -xdim, -ydim, -topol, and -neigh
#      parameters if needed.
#      Orwig creates an array of unit vectors
#      (unit190.vct) which is used
#      in "reverse mode" to determine an
#      assignment of the "best feature"
#      to each node of the Kohonen grid.
example :
./randinit -din ex.dat -cout ex.cod -xdim 10 -ydim 20 -topol hexa \
-neigh bubble -rand 1 -v 2

./vsom      -din ex.dat -cin ex.cod -cout ex.cod -rlen 1000 \
-alpha 0.05 -radius 10 -rand 1 -v 2

./vsom      -din ex.dat -cin ex.cod -cout ex.cod -rlen 10000 \
-alpha 0.02 -radius 3 -rand 1 -v 2

./qerror    -din      ex.dat -cin ex.cod -v 2
#      Map is created as ex.cod

#      Derive the positions of the comments by
#      passing comment file
#      against the trained grid:
./visual    -din ex.dat -cin ex.cod -dout comments.cmt

#      Derive the values for each node of the
#      grid by passing the grid
#      against a unit vector list:
./visual    -din ex.cod -cin unit190.vct -dout map.ids

#      Derive the positions of the unit vector of
#      terms:
./visual    -din unit190.vct -cin ex.cod -dout terms.trm

```


REFERENCES

- Agosti, M., Crestani, F., & Gradenigo, G. (1989). Towards data modelling in information retrieval. *Journal of Information Science*, 15, 307-319.
- Alterman, R. (1991). Understanding and summarization. *Artificial Intelligence Review*, 5, 239-254.
- Anderson, J. R., Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (1983). *Machine Learning: an artificial intelligence approach*. Palo Alto, CA: Tioga Publishing Company.
- Andrew, A. M. (1989). *Self-organizing Systems*. New York: Gordon and Breach Science Publishers.
- Argyris, C., & Schon, D. A. (1978). *Organizational Learning: A Theory of Action Perspective*. Reading, MA: Addison-Wesley.
- Babloyantz, A. (1989). *Self-Organization, Emerging Properties, and Learning*. New York: Plenum Press.
- Barrett, E. (1988). *Text, ConText, and HyperText: Writing with and for the Computer*. Cambridge, Massachusetts: The MIT Press.
- Belew, R. K. (1989). Adaptive information retrieval. *Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Cambridge, MA, 11-20.
- Belkin, N. J. (1990). The cognitive viewpoint in information science. *Journal of Information Science*, 16, 11-15.
- Belkin, N. J., & Croft, W. B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Communications of the ACM*, 35(12 (December)), 29-38.
- Blair, D. C. (1990). *Language and Representation in Information Retrieval*. Amsterdam: Elsevier.
- Blum, A. (1992). *Neural Networks In C++: An Object-Oriented Framework for Building Connectionist Systems*. New York: John Wiley & Sons, Inc.

- Booker, L. B., Goldberg, D. E., & Holland, J. H. (1989). *Classifier Systems and Genetic Algorithms*. Morgan Kaufmann Publishers, Inc.
- Boulding, K. E. (1956). *General Systems Theory -- The Skeleton of Science*. Paper presented at the Management Science.
- Brachman, R. J. (1979). *On the Epistemological Status of Semantic Networks*. Academic Press.
- Brookes, B. C. (1977). *The developing cognitive viewpoint in information science*. Ghent.
- Burrell, G., & Morgan, G. (1979). *Social paradigms and organizational analysis*. London: Heinemann Educational Books.
- Burton, R. R. (1976). *Semantic grammar: A technique for efficient language understanding in limited domains*. Unpublished Doctoral Dissertation, Computer Science Department, University of California at Irvine.
- Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ 07642: Lawrence Erlbaum Associates, Inc.
- Caudill, M. (1989). *Using Neural Nets: Representing Knowledge*. Paper presented at the AI Expert.
- Caudill, M. (1991). Neural Network Training Tips and Techniques. *AI Expert*, pp. 56-61.
- Caudill, M. (1993). A Little Knowledge is a Dangerous Thing. *AI Expert*, 8(6 (June)), 16-22.
- Chen, H. (1994). Collaborative Systems: Solving the Vocabulary Problem. *IEEE Computer*, 27(5), 58-66.
- Chen, H., Basu, K., & Ng, T. (1992). Concept exploration in large knowledge networks: symbolic branch-and-bound vs. connectionist Hopfield net activation. *CMI-WPS*, 92-11.
- Chen, H., & Dhar, V. (1991). Cognitive process as a basis for intelligent retrieval systems design. *Information Processing and Management*, 27(5), 405-432.

- Chen, H., Hsu, P., Orwig, R., Hoopes, L., & Nunamaker, J. F., Jr. (1994). Automatic Concept Classification of Text from Electronic Meetings. *Communications of the ACM*, 37(10).
- Chen, H., & Lynch, K. J. (1992). Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man and Cybernetics*, 22(5 (September/October)), 885-902.
- Chen, H., Lynch, K. J., Basu, K., & Ng, T. (1993). Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE Expert*, 8(2 (April)), 25-34.
- Chen, H., Orwig, R., Ng, K., Martinez, J., Sutjahjo, S., & Nunamaker, J. F., Jr. (under review). An Intelligent Agent for Electronic Meetings.
- Chiang, R. H. L. (1991). *Extraction of Conceptual Models from Relational Databases*. Paper presented at the Proceedings of the First Workshop on Information Technologies and Systems (WITS'91), Cambridge, MA.
- Chisholm, R. M. (1966). *Theory of Knowledge*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Chomsky, Noam. (1965). *Aspects of the theory of syntax*. Cambridge, MA: M.I.T. Press.
- Connolly, T., Jessup, L., & Valacich, J. (1990). Effects of Anonymity and Evaluative Tone on Idea Generation in Computer-Mediated Groups. *Management Science*, 36(6), 689-703.
- Crouch, C. J. (1990). An Approach to the Automatic Construction of Global Thesauri. *Information Processing & Management*, 26(5), 629-640.
- Daft, R. L., & Weick, K. E. (1984). Toward a Model of Organizations as Interpretation Systems. *Academy of Management Review*, 9(2), 284-294.
- Dalenoort, G. J. (1989). *The Paradigm of Self-Organization*. New York: Gordon and Breach Science Publishers.
- Dalton, J., & Deshmene, A. (1991). Artificial neural networks: An approach to increasing machine intelligence. *IEEE Potentials*, 4(33 (April)), 33-36.

- Date, C. J. (1990). *An Introduction to Database Systems Volume I*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- de Amorim, S. G., Barthelemy, J.-P., & Ribeiro, C. C. (1992). Clustering and Clique Partitioning: Simulated Annealing and Tabu Search Approaches. *Journal of Classification*, 9, 17-41.
- de Mey, M. (1977). *The cognitive viewpoint: its development and its scope*. Ghent.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6 (Sept.)), 391-407.
- Dennis, A. R., George, J. F., M., J. L., Nunamaker, J. F., Jr., & Vogel, D. R. (1988). Information Technology to Support Electronic Meetings. *Management Information Systems Quarterly*, 12(4), 591-624.
- Dennis, A. R., Nunamaker, J. F., Jr., & Vogel, D. (1991). A comparison of laboratory and field research in the study of electronic meeting systems. *Journal of Management Information Systems*, 7(3), 107-135.
- Dennis, A. R., Valacich, J., & Nunamaker, J. F., Jr. (1991). *Group, Sub-Group and Nominal Group Idea Generation in an Electronic Meeting Environment*. Paper presented at the 24th Hawaii International Conference on System Sciences, Hawaii.
- Doszkocs, T. E., Reggia, J., & Lin, X. (1990). Connectionist models and information retrieval. *Annual Review of Information Science and Technology (ARIST)*, 25, 209-260.
- Ebeling, W., Peschel, M., & Weidlich, W. (1991). *Models of self-organization in complex systems: MOSES*. Berlin: Akademie Verlag.
- Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1 January), 38-58.
- Ericsson, K. A., & Simon, H. A. (1984). *Protocol Analysis: Verbal Reports as Data*. Cambridge, Massachusetts: The MIT Press.
- Everitt, B. (1980). *Cluster Analysis*. (Second ed.). London: Heinemann Educational Books.

- Faraj, N., Godin, R., Missaoui, R., David, S., & Plante, P. (1994). Evaluation of the contribution of syntactic composite terms to automatic indexing. *CAIS/ACSI '94 Annual Proceedings -- Information Industry in Transition*, Montreal, Que., 25-27 May 1994, 340-60.
- Fielding, N. G., & Lee, R. M. (1991). *Using Computers in Qualitative Research*. London: Sage Publications Ltd.
- Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms (Ed.), *Universals in Linguistic Theory*. New York, NY: Holt, Rinehart, and Winston.
- Foltz, P. W., & Dumais, S. T. (1992). Personalized Information Delivery: An Analysis of Information Filtering Methods. *Communications of the ACM*, 35(12 (December)), 51-70.
- Fox, William M., (1989). The improved Nominal Group Technique (NGT). *Journal of Management Development*, 8(1), 20-27.
- Frakes, W. B., & Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, NJ 07632: Prentice Hall.
- Frankel, Stanley, (1987). NGT + MDS: An adaptation of the nominal group technique for ill-structured problems. *Journal of Applied Behavioral Science*, 23(4), 543-551.
- Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1991). *Knowledge Discovery in Databases: An Overview*. AAAI Press.
- Furnas, G. W. (1992). Envisioning information - tuft, er. *Journal of classification*, 9(1), 154-158.
- Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., & Lochbaum, K. E. (1988). *Information retrieval using a singular value decomposition model of latent semantic structure*. Paper presented at the 11th International Conference on Research and Development in Information Retrieval (13-15 June), Grenoble, France.
- Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal*, 62(6 (Jul-Aug, Pt 3)), 1753-1806.

- Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11 (November)), 964-971.
- Furnas, G. W., Robertson, S. P., Olson, G. M., & Olson, J. S. (1991). *New graphical reasoning models for understanding graphical interfaces*. Paper presented at the Human Factors in Computing Systems. Reaching Through Technology., New Orleans, LA, USA.
- Gaines, B. R. (1991). *The Trade-Off Between Knowledge and Data In Knowledge Acquisition*. AAAI Press.
- George, J. F., Easton, G. K., Nunamaker, J. F., Jr., & Northcraft, G. B. (1990). A study of collaborative group work with and without computer-based support. *Information Systems Research*, 1(4 December), 394-415.
- George, J. F., Nunamaker, J. F., Jr., & Valacich, J. S. (1992). Electronic meeting systems as innovation: a study of the innovation process. *Information and Management*, 22(3 (March)), 187-95.
- Gersho, M., & Reiter, R. (1990a). *Information Retrieval using a Hybrid Multi-Layer Neural Network*. Paper presented at the Proceedings of the IJCNN, San Diego.
- Gersho, M., & Reiter, R. (1990b). *Information Retrieval using Self-Organizing and Heteroassociative Supervised Neural Network*. Paper presented at the Proceedings of the INNC, Paris.
- Ginsberg, A. (1993). A unified approach to automatic-indexing and information-retrieval. *IEEE Expert*, 8(5), 46-56.
- Giunchiglia, F., & Walsh, T. (1992). A theory of abstraction. *Artificial Intelligence*, 57, 323-389.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.
- Greif, I. (1994). Desktop Agents in Group-Enabled Products. *Communications of the ACM*, 37(7 (July)), 100-105.

- Grohowski, R. B., McGoff, C., Vogel, D. R., Martz, W. B., & Nunamaker, J. F., Jr. (1990). Implementing electronic meeting systems at IBM: Lessons learned and success factors. *Management Information Systems Quarterly*, 14(4), 369-383.
- Haken, H. (1988). *Information and self-organization: a macroscopic approach to complex system*. Berlin: Springer-Verlag.
- Han, J., Cai, Y., & Cercone, N. (1993). Data-Driven Discovery of Quantitative Rules in Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(1 (February)), 29-40.
- Harris, M. D. (1985). *Introduction to Natural Language Processing*. Reston, Virginia: Reston Publishing Company, Inc.
- Herden, G. (1990). *On Cluster Methods for Qualitative Data II*. Springer-Verlag.
- Hiotis, A. (1993). Inside a Self-Organizing Map. *AI Expert*, 8(4 (April)), 38-43.
- Hopfield, J. J. (1982). Neural network and physical systems with collective computational abilities. *Proceedings of the National Academy of Science*, 78(8).
- Huber, G. P. (1988). Effects of decision and communication technologies on organizational decision processes and structures. In R. M. Lee, A. McCosh, & P. Migliarese (Ed.), *Organizational Decision Support Systems*(pp. 317-333). Amsterdam: North Holland.
- Huber, G. P. (1990). A Theory of the Effects of Advanced Information Technologies on Organizational Design, Intelligence, and Decision Making. *Academy of Management Review*, 15(1).
- Jacobson, T. L. (1991). Sense-Making in a Database Environment. *Information Processing & Management*, 27(6), 647-657.
- James, M. (1985). *Classification Algorithms*. London: Collins Professional and Technical Books.
- Jessup, L., & Tansik, D. (1991). Decision Making in an Automated Environment: The Effects of Anonymity and Proximity on Group Process and Outcome with a Group Decision Support System. *Decision Sciences*, 22(1), 266-279.

- Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. New York, NY: The Free Press.
- Johnson, J. J. (1993). *Supporting Decision Making with Organizational Decision Memory*. Unpublished Doctoral Dissertation, University of Arizona.
- Jones, K. S. (1971). *Automatic Keyword Classification for Information Retrieval*. London: Butterworths.
- Jones, K. S. (1991). Notes and references on early automatic classification work. *ACM Special Interest Group on Information Retrieval (SIGIR) Forum*, 25(1), 10-17.
- Kaufman, K. A., Michalski, R. S., & Kerschberg, L. (1991). *Mining for Knowledge in Databases: Goals and General Description of the INLEN System*. AAAI Press.
- Kirkpatrick, D. (1992, March 23). Here comes the payoff from PCs. *Fortune*, pp. 93-100.
- Knight, K. (1990). Connectionist ideas and algorithms. *Communications of the CACM*, 33(11), 59-74.
- Kochen, M. (1989). Management Intelligence Systems. *Advances in Computers*, 28, 227-278.
- Kohonen, T. (1989). *Self-Organization and Associative Memory*. Berlin: Springer-Verlag.
- Kwok, K. L., (1989) A neural network for probabilistic information retrieval. *Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Cambridge, MA, 21-30.
- Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. Chicago, IL 60637: The University of Chicago Press.
- Larson, R. R. (1991). Classification Clustering, Probabilistic Information Retrieval, and the Online Catalog. *The Library Quarterly*, 61(2 (April)), 133-173.
- Larson, R. R. (1992). Experiments in Automatic Library of Congress Classification. *Journal of the American Society for Information Science*, 43(2 (March)), 130-148.

- Latta, G. F., & Swigger, K. (1992). Validation of the Repertory Grid for Use in Modeling Knowledge. *Journal of the American Society for Information Science*, 43(2 (March)), 115-129.
- Lehnert, W., & Sundheim, B. (1991). A Performance Evaluation of Text-Analysis Technologies. *AI Magazine*, pp. 81-94.
- Lewis, C. M., & Sycara, K. P. (1993). Reaching Informed Agreement in Multispecialist Cooperation. *Group Decision and Negotiation*, September, 279-300.
- Lin, X., Soergei, D., & Marchionini, G. (1991). *A Self-organizing Semantic Map for Information Retrieval*. Paper presented at the Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Chicago, IL USA.
- Lippmann, R. P. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, pp. 4-22.
- Luksch, P., & Wille, R. (1990). *A Mathematical Model for Conceptual Knowledge Systems*. Springer-Verlag.
- MacLeod, K. J., & Robertson, W. (1991). A Neural Algorithm for Document Clustering. *Information Processing & Management*, 27(4), 337-346.
- Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., & Cohen, M. D. (1987). Intelligent Information-Sharing Systems. *Communications of the ACM*, 30(5 (May)), 390-402.
- Marik, K. (1989). *Knowledge representation and organization in machine learning*. Berlin: Springer-Verlag.
- Martz, W. B., Jr., Vogel, D. R., & Nunamaker, J. F., Jr. (1992). Electronic meeting systems: results from the field. *Decision Support Systems*, 8(2 (April)), 141-58.
- McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Boston, Massachusetts: The MIT Press.
- McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in neural nets. *Bulleting of Mathematical Biophysics*, 5, 115-137.

- McKevitt, P., Partridge, D., & Wilks, Y. (1992). Approaches to natural language discourse processing. *Artificial Intelligence Review*, 6, 333-364.
- Merkl, D., Tjoa, A. M., & Kappel, G. (1993). *Retrieval of Reusable Software Based on Semantic Similarity: An Artificial Neural Network Approach*. Institut für Angewandte Informatik und Informationssysteme, Universität Wien.
- Merkl, D., Tjoa, A. M., & Kappel, G. (1994). *A Self-Organizing Map that Learns the Semantic Similarity of Reusable Software Components*. Paper presented at the Proceedings of 5th Australian Conference on Neural Networks, Brisbane.
- Messatfa, H. (1992). An Algorithm to Maximize the Agreement Between Partitions. *Journal of Classification*, 9, 5-15.
- Michalski, R., & Stepp, R. (1983). Learning from observation: conceptual clustering. In R. Michalski, R. Carbonell, & J. Mitchell (Ed.), *Machine Learning, An Artificial Intelligence Approach* (pp. 331-363). Palo Alto, CA: Tioga Publishing Company.
- Michalski, R. S. (1983). *A theory and methodology of inductive learning*. Tioga Publishing Company.
- Michalski, R. S. (1986). *Understanding the Nature of Learning: Issues and Research Directions*. Morgan Kaufmann.
- Miikkulainen, R. (1993). *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. Cambridge, MA: The MIT Press.
- Morgan, M. (1990). *Machine Learning: Proceedings of the Seventh International Conference (1990)*. Palo Alto, CA: Morgan Kaufmann Publishers, Inc.
- Morik, K. (1987). *Knowledge Representation and Organization in Machine Learning*. Berlin: Springer-Verlag.
- Morrison, J. (1992). *Development and Evaluation of a System to Support Team and Organizational Memory*. Unpublished Doctoral Dissertation, University of Arizona.
- Müller, A., & Neumann, J. (1990). *Classification with neural networks*. Springer-Verlag.

- Newell, A., & Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice Hall, Inc.
- Nunamaker, J. F., Jr. (1992, December, 1992). Corporate Consciousness: In Search of Group Memory. *Corporate Computing*.
- Nunamaker, J. F., Jr., Chen, M., & Purdin, T. D. M. (1991). Systems Development in Information Systems Research. *Journal of Management Information Systems*, 7(3 (Winter)), 89-106.
- Nunamaker, J. F., Jr., Dennis, A. R., Valacich, J. S., & Vogel, D. R. (1991). Information technology for negotiating groups: generating options for mutual gain. *Management Science*, 37(10), 1325-1346.
- Nunamaker, J. F., Jr., Dennis, A. R., Valacich, J. S., Vogel, D. R., & George, J. F. (1991). Electronic Meeting Systems to Support Group Work. *Communications of the ACM*, 34(7 (July)), 40-61.
- Poole, M. S., & Jackson, M. (1993). Communication Theory and Group Support Systems. In L. Jessup & J. Valacich (Ed.), *Group Support Systems: New Perspectives*(pp. 281-294): MacMillan.
- Post, B. (1992). *Building the business case for group support technology*. Paper presented at the 25th Hawaii International Conference on System Sciences, Hawaii.
- Quinlan, J. R. (1983). *Learning Efficient Classification Procedures and Their Application to Chess End Games*. Morgan Kaufmann Publishers, Inc.
- Rada, R. (1991). *Hypertext: from Text to Expertext*. London: McGraw-Hill Book Company.
- Rahmstorf, G. (1990). *Compositional Semantics and Concept Representation*. Springer-Verlag.
- Rama, D. V., & Srinivasan, P. (1993). An Investigation of Content Representation Using Text Grammars. *ACM Transactions on Information Systems*, 11(1 (January)), 51-75.
- Rasmussen, E. (1992). *Clustering Algorithms*. Prentice Hall.

- Rich, E., & Knight, K. (1991). *Artificial Intelligence*. New York: McGraw-Hill, Inc.
- Ritter, H., & Kohonen, T. (1989). Self-Organizing Semantic Maps. *Biological Cybernetics*, 61, 241-254.
- Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: An Introduction*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning internal representations by error propagation*. The MIT Press.
- Sager, N., Friedman, C., & Lyman, M. S. M. (1987). *Medical Language Processing: Computer Management of Narrative Data*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- Salmon, W. C. (1967). *The Foundations of Scientific Inference*. Pittsburgh, PA 15260: University of Pittsburgh Press.
- Salton, G. (1978). Generation and search of clustered files. *ACM Transactions on Database Systems*, 3(4), 321-346.
- Salton, G. (1989). *Automatic Text Processing*. Reading, MA: Addison-Wesley Publishing Company, Inc.
- Sandoe, K., Olfman, L., & Mandviwalla, M. (1991). *Meeting in Time: Recording the Workgroup Conversation*. Paper presented at the International Conference on Information Systems, New York.
- Scholtes, J. C. (1993). *Neural Networks in Natural Language Processing and Information Retrieval*. Unpublished Doctoral Dissertation, Universiteit van Amsterdam.
- Shekar, B., Murty, M. N., & Krishna, G. (1987). *Pattern Clustering: An Artificial Intelligence Approach*. Paper presented at the Proceedings of the Tenth International Joint Conference on Artificial Intelligence.
- Shulldberg, H. K., Macpherson, M., Humphrey, P., & Corley, J. (1993). Distilling Information from Text: The EDS TemplateFiller System. *Journal of the American Society for Information Science*, 44(9 (October)), 493-507.

- Simpson, P. (1990). *Artificial Neural Systems: foundations, Paradigms, Applications, and Implementations*. New York, NY: McGraw-Hill Book Company.
- Smyth, P., & Goodman, R. M. (1992). An Information Theoretic Approach to Rule Induction from Databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4 (August)), 301-316.
- Sparck-Jones, K. (1971). *Automatic Keyword Classification for Information Retrieval*. London: butterworths.
- Sparck-Jones, K. (1991). *Notes and references on early automatic classification work*. Paper presented at the ACM Special Interest Group on Information Retrieval (SIGIR) Forum.
- Sriram, N. (1990). Clique Optimization: A Method to Construct Parsimonious Ultrametric Trees from Similarity Data. *Journal of Classification*, 7, 33-52.
- Stepp, R. E. (1987). *Concepts in Conceptual Clustering*. Paper presented at the Proceedings of the Tenth International Joint Conference on Artificial Intelligence.
- Strauss, A., & Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA 91320: Sage Publications, Inc.
- Tanaka, J. W., & Taylor, M. (1991). Object categories and expertise: Is the basic level in the eye of the beholder? *Cognitive Psychology*, 23(3 (July)), 457-482.
- Tank, D., & Hopfield, J. (1987, December). Collective computation in neuronlike circuits. *Scientific American*, pp. 104-114.
- Tesch, R. (1990). *Qualitative Research: Analysis Types and Software Tools*. Bristol, PA 19007: The Falmer Press.
- Valacich, J., Dennis, A., & Nunamaker, J. F., Jr. (1992). Group size and anonymity effects on computer-mediated idea generation. *Small Group Research*, 23(1), 49-73.
- Valacich, J. S., Dennis, A. R., & Nunamaker, J. F., Jr. (1991). Electronic meeting support: the GroupSystems concept. *International Journal of Man-Machine Studies*, 34(2 (Feb.)), 261-82.

- Valacich, J. S., Vogel, D., & Nunamaker, J. F., Jr. (1989). *Integrating Information Across Sessions and Between Groups in Group Decision Support Systems*. Paper presented at the Proceedings of the 22nd Annual Hawaii International Conference on System Sciences, Kailua-Kona, Hawaii.
- Vogel, D. R., & Nunamaker, J. F., Jr. (1990). Group decision support system impact: Multi-methodological exploration. *Information and Management, Winter*, 15-28.
- Vogel, D. R., Nunamaker, J. F. J., Martz, W. B. J., Grohowski, R., & McGoff, C. (1989). Electronic Meeting System Experience at IBM. *Journal of Management Information Systems*, 6(3 (Winter)), 25-43.
- Vogt, F., Wachter, C., & Wille, R. (1990). *Data Analysis Based on a Conceptual File*. Springer-Verlag.
- Weiss, S. M., & Kulikowski, C. A. (1991). *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA 94403: Morgan Kaufmann Publishers, Inc.
- Wilson, T. D. (1984). The cognitive approach to information-seeking behavior and information use. *Social Science Information Studies*, 4, 197-204.
- Wong, S. K. M., & Yao, Y. Y. (1992). An Information-Theoretic Measure of Term Specificity. *Journal of the American Society for Information Science*, 43(1), 54-61.
- Woods, W. A. (1972). An experimental parsing system for transition network grammars. In R. Rustin (Ed.), *Natural Language Processing*. New York, NY: Algorithmics Press.
- Ziarko, W. (1991). *The Discovery, Analysis, and Representation of Data Dependencies in Databases*. AAAI Press.