Volume 24 Number 2 June 2000 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Group Support Systems

Guest Editors:

Gary Klein,

Morgan M. Shepherd

Informatica **24** (2000) Number 2, pp. 147–284

The Slovene Society Informatika, Ljubljana, Slovenia

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: http://, Europe: http://ai.ijs.si/informatica, Asia: http://www.comp.nus.edu.sg/ liuh/Informatica/index.html.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2000 (Volume 24) is
– DEM 100 (US$ 70) for institutions,
– DEM 50 (US$ 34) for individuals, and
– DEM 20 (US$ 14) for students
plus the mail charge DEM 10 (US$ 7).
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, Kranj, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 61 1773 900, Fax (+386) 61 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Janez Peklenik)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Engineering Index, INSPEC, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik, Linguistics and Language Behaviour Abstracts, Cybernetica Newsletter

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102 Ljubljana. Slovenia taxe Percue.

Time Pressure Impacts on Electronic Brainstorming in a Group Support System Environment

Jay E. Aronson
Department of Management Information Systems, Terry College of Business, The University of Georgia, Athens, USA
Phone: 706 542-0991, Fax: 706 583-0037
E-mail: jaronson@terry.uga.edu
AND
Robert M. Myers and Robert B. Wharton
Rinker School of Business, Palm Beach Atlantic College, West Palm Beach, Florida, USA
Phone: 561 803-2462, 803-1469, Fax: 561 803-2467
E-mail: myerssr@pbac.edu

Keywords: Group Support Systems, Electronic Brainstorming, Electronic Meeetings, Collaborative Computing

Edited by: Morgan Shepherd
Received: June 11, 1999 **Revised:** October 10, 1999 **Accepted:** December 9, 1999

One important task in the decision making process in an organization is that of brainstorming. In a Group Support System (GSS) environment, electronic brainstorming is enabled by special hardware and software that introduces both task gains and losses; and process gains and losses. The impacts of time pressure on task completion and task quality are important concerns for managers. Specifically, we focus on the impacts of time pressure on electronic brainstorming in terms of idea quantity per unit time (idea generation rate) and idea quality (creativity). One hundred and two undergraduate business students were subjects in an experiment examining the impacts of time pressure on: (a) the rate of ideas generated; and (b) the quality (creativity) of the generated ideas. In the treatments, we varied the amount of time available for working on an electronic brainstorming task over time. Social Entrainment Theory indicates that there will be a lasting effect as the length of time to perform a task over several treatments is varied. We hypothesize that the impacts explained by Social Entrainment Theory will occur in the GSS setting. Our results support the research hypotheses that the mean rate of: (a) idea generation, and (b) the creativity of the ideas is unequal in groups operating under different time pressure conditions.

1 Introduction

Many organizations want to improve group decision making activities. Consequently, there are volumes of research that identify methods to produce more efficient and effective group decision making. One important task in the decision making process in an organization is that of brainstorming. In a Group Support System (GSS) environment, electronic brainstorming is enabled by special hardware and software that introduce both task gains and losses; and process gains and losses (Turban and Aronson 1998). Generally, one expects overall gains in task performance and enhancements to processes due to the introduction of special technologies designed to enhance group work. However, this does not always occur. Sometimes task and process losses may be introduced. For example, one potential task loss in a GSS electronic brainstorming session is information overload, when more ideas are generated than can be 'digested' and utilized by meeting participants in a reasonable time frame. For other examples of gains and losses, see Nunamaker et al. (1993) and Turban and Aronson (1998). Collaborative computing efforts by workgroups are enhanced by new forms of groupware (software and hardware that enables collaborative computing), such as GroupSystems for Windows (Ventana Corporation) and Lotus Notes / Domino Server (Lotus Development Corp.). These are diffusing throughout organizations for use in collaborative work, and for distance learning, an area into which many universities and colleges are moving along with corporations (see Reinig, Briggs, Brandt and Nunamaker, 1997).

It is becoming increasingly important for a decision maker to determine not only the appropriate sequence of tasks required for a workgroup, but also to allocate the right amount of time to each task to 'optimize' the group's performance in terms of quantity and quality of work. Decision making under time pressure is a critical issue in the modern organization. Since information is available at the touch of a button, it must be synthesized and utilized quickly to become and remain competitive. Practically speaking, managers impose appropriate deadlines for task completion in the workplace. There are tradeoffs in

the quantity of work performed and its quality under time pressure. Given less time, quality and quantity tends to decrease. Given too much time, quality tends to decrease, while quantity may increase. A rushed decision maker may not make as good a decision that he/she could have made given more time. In the modern enterprise, people often work in groups. Likewise, groups also experience time pressures, and prior research on group work indicates that both the quantity and quality (a.k.a. creativity) of a group's output is directly affected by time pressure. Given too much time or too little time, and a group will underperform in certain ways.

Much of the early research on collaborative group work did not involve the use of technology but made use of other techniques often centering on the concept of brainstorming. Osborn (1957) devised brainstorming theory as a method of group problem solving to increase the quality and quantity of ideas developed by group members. In today's fast-paced business environment, time pressure is a major factor in conducting business. However, there have been few research studies that have investigated time pressure as it relates to group processes, productivity, and creativity (Kelly and McGrath, 1985).

Some time pressure studies have found that group decisions made under time pressure have lead to poor performance for a variety of groups including government committees (Janis, 1982), various businesses (Thurow, 1980), and even juries (Greenberg, Williams and O'Brien, 1986). Providing too much time for group decisions may even lead to boredom and dissatisfaction (Karau and Kelly, 1993). In a GSS setting, this has been observed as a 'popcorn' effect: that initially, ideas are generated quickly, they peak, and then taper off to the point where only the last few subjects are typing. The remainder of the subjects are either reading or waiting. If too much time is allocated for a brainstorming task, the typing sounds like a bag of microwave popcorn cooking during the last minute or so (e.g., see Myers, 1997). Therefore, the selection of the proper time frame for groups to complete tasks successfully is critical.

Specifically, we focus on the impacts of time pressure on electronic brainstorming in terms of idea quantity per unit time (idea generation rate) and idea quality (creativity). One hundred and two undergraduate business students were subjects in an experiment examining the impacts of time pressure on: (a) the rate of ideas generated; and (b) the quality (creativity) of the generated ideas. In the treatments, we varied the amount of time available for working on an electronic brainstorming task over time. Social Entrainment Theory indicates that there will be a lasting effect as the length of time to perform a task over several treatments is varied. We hypothesize that the impacts explained by Social Entrainment Theory will occur in the GSS setting.

Our results support the research hypotheses that the mean rate of: (a) idea generation, and (b) the creativity of the ideas is unequal in groups operating under different time pressure conditions.

This paper is organized as follows. In the next section, we discuss the definitions of GSS and related concepts. Following this, we discuss the theory and early research on which this effort is based. We also propose two hypotheses concerning the impacts of time pressure on electronic brainstorming in a GSS setting. In the following section, we describe our method. The last four sections consist of results, discussion, limitations and future research, and implications for GSS researchers and practitioners.

2 Group Support Systems

A Group Decision Support System (GDSS) is an interactive computer-based system that facilitates the solution of unstructured problems by a group of decision makers (DeSanctis and Gallupe, 1987). Components include hardware, software, people, and procedures that support the process of group work (e.g., see Turban and Aronson, 1998). In the 1990s, researchers eliminated the D from the GDSS acronym to generalize the term to Group Support System (GSS), because they determined that most of the systems were used to enhance meeting performance, but did not necessarily lead to a decision. The terms GSS and Electronic Meeting System (EMS) are used synonymously. An EMS is

> An information technology (IT)-based environment that supports group meetings, which may be distributed geographically and temporally. The IT environment includes, but is not limited to, distributed facilities, computer hardware and software, audio and video technology, procedures, methodologies, facilitation, and applicable group data. Group tasks include, but are not limited to communication, planning, idea generation, problem solving, issue discussion, negotiation, conflict resolution, system analysis and design, and collaborative group activities such as document preparation and sharing (Dennis et al., 1988).

A decision room is a special facility containing computer workstations and a large public screen. The GSS software allows the participants to interact through a variety of tools including electronic brainstorming and ranking / voting. Our study focuses on electronic brainstorming (i.e., idea generation), a common task in GSS settings. Next we discuss specifics of the base theory and research.

3 Theory and Research

Concepts associated with brainstorming have been widely investigated since Osborn (1957) first proposed the theory. Over twenty studies have found that nominal groups (individuals generating ideas on their own which are then combined with the ideas of other individuals also working on their own) generate more ideas than the same number of

people in face-to-face interacting groups (Mullen, Johnson and Salas, 1991; Gallupe, Bastianutti, and Cooper, 1991). Consequently, researchers have attempted to create environments that enhance group performance.

Only a few studies have examined the impact of time constraints on group productivity. A variety of approaches describe how time constraints influence group productivity. Wright (1974) found that group members under time pressure systematically place greater weight on negative evidence than those group members not under time pressure. Zakay and Wooler (1984) describe how high effectiveness diminished rapidly when time pressure was applied. Under time pressure, old habits that tend to overrule newly learned skills are activated. The reemergence of old habits may be deemed a form of social entrainment where an external condition (such as a time limit) alters the way the group is working. We turn to this concept next.

3.1 Time Pressure and Entrainment

The entrainment concept relating to group performance is a relatively new area of research. Kelly and McGrath (1985) initially investigated the entrainment concept applied to group performance. They defined social entrainment as "a concept that refers to the altering of social rhythms or patterns by external conditions (such as time limits), and to the persistence of such new rhythms over time" (p. 395). This concept may have direct impacts on business in terms of optimizing training programs. However, our interest focuses directly on how time limits alter work patterns. We are interested in establishing how the quantity and quality of ideas in an electronic brainstorming task are altered by the concept of entrainment. Given an understanding of how entrainment impacts on task performance, a good manager can determine the correct sequence of tasks, along with appropriate time allocations for each one. Brainstorming is one task that decision makers perform on a regular basis. Though electronic group meetings have steps that follow brainstorming, we focus on this first one. Later studies may be conducted to establish the impacts of time pressure on the entire electronic meeting process.

Their 1985 study of entrainment utilized 512 undergraduate students. The students were given two tasks consisting of either production, planning, or discussion tasks as identified by Hackman (1966). Production tasks required the group to generate ideas. Discussion tasks required the group to evaluate an issue and planning tasks required the group to describe a plan of action to achieve a goal (Kelly and McGrath, 1985). Dimensions that assessed both quality and quantity of idea generation were selected from those developed by Hackman, Jones, and McGrath (1967). The groups were given two time periods in which to work. One time period was 10 minutes while the second time period was 20 minutes. The time periods were varied between groups (Kelly and McGrath, 1985).

At the conclusion of the experiments, the group members were asked to identify the main source of stress they experienced during the experiments. This assessment was completed through the use of post-test questionnaires. The format of the data collection was free response. All responses were then classified into one of two categories: (a) time pressure, or (b) all other responses (Kelly and McGrath, 1985).

This study also utilized two judges rating each of the dimensions utilized in the research. The judges used a 7-pile sort-resort technique as noted by Hackman, Jones and McGrath (1967). As part of the research, the inter-rater reliability of the judges was assessed (Kelly and McGrath, 1985). This study found that time constraints do influence group performance. The shorter time period led to higher rates of performance but at a cost to quality. The 20 minute time period generated higher quality ideas than those generated by the 10 minutes period groups (Kelly and McGrath, 1985). This study also supported the concept of social entrainment. Kelly and McGrath (1985) found that persistence of interaction and performance patterns continued even when the situational conditions (time pressure) was altered.

Kelly, Futoran, and McGrath (1990) continued the research by reporting the results of seven studies that investigated the entrainment concept. Their method was similar to the previous study by Kelly and McGrath (1985). All of the studies utilized undergraduate students who received course credit for their participation. The various studies used groups who operated under differing time periods. These periods were varied among the groups.

The tasks used in these studies included the unusual-uses tasks[1] that were identified by McGrath (1984). These tasks required groups to develop unusual-uses for common items. For each of these time trials the groups consisted of either dyads or triads. The time constraints for these seven experiments varied between 5 minutes and 20 minutes depending upon the task used (Kelly, Futoran and McGrath, 1990).

This study found that entrainment effects can be divided into two types: (a) initial trial effects, or (b) trial-to-trial carry-over. This study found that when investigating rate of ideas, initial trial effects show that short initial trials lead to faster rates of performance while long initial trials lead to slower rates of performance. The trial-to-trial carry-over effects on rate of idea generation showed groups that ex-

[1]Unusual-tasks problems involve brainstorming efforts on very uncommon issues that most people typically would not have encountered before. These are useful for comparing treatments in a GSS experimental setting for three reasons: firstly, the subject is unlikely to have encountered the task previously; secondly, the subject can easily understand the task because the function and content of the object in question is familiar; and thirdly, the task has an appropriate level of complexity so as to make it interesting, somewhat challenging, and yet quite doable by the subject. Examples of unusual-uses tasks include 1) the two thumbs problem ("What additional tasks would you be able to perform if you had two thumbs on each hand?"); 2) the paperclip problem ("What can be done with a production overrun of paperclip wire material?"); and 3) the tea bag problem ("What can be done with an overrun of teabag netting?"). For others, see Wheeler, B. and B. Mennecke (eds.), "Research Tasks Repository," at ISWorld Net, http://ais-notes.bus.indiana.edu/isworld/tasks.nsf, and follow the links from "Group Tasks by Type" to "Idea Generation."

perienced difficulty as related to capability seemed to slow down their rate on subsequent trials. Groups that experienced difficulty as related to capacity seemed to speed up rates on later trials (Kelly, Futoran and McGrath, 1990).

Kelly, Futoran and McGrath (1990) discussed these concepts of capacity and capability. Capacity problems generally involve issues of either time, load, or difficulty level. If a group is facing capacity problems associated with time pressure, they will attempt to compensate for the capacity problem by speeding up their rate. The concept of entrainment is observed when the group continues this pace in a subsequent trial even when the time limit has been relaxed (Kelly, Futoran and McGrath, 1990). Kelly, Futoran and McGrath (1990) noted that groups operating under classic brainstorming instruction would most likely experience capacity problems as they attempt to obtain as many ideas as possible.

Problems associated with capability are generally defined as tasks that are "beyond the unit's current or momentary task performance capabilities" (p. 287). The method groups use to deal with this problem is not to speed up their work but rather to slow down. This slow down allows the group more processing time for the task (Kelly, Futoran and McGrath, 1990).

Kelly, Futoran and McGrath (1990) investigated how the concepts of capacity and capability relate to time pressure and group decision making. Capacity problems generally involve issues of either time, load, or difficulty level. They discovered that if a group faces time pressure-based capacity problems, the members attempt to compensate by speeding up their work rate. (We base our treatments in which we measure the rate of idea generation on Kelly, Futoran and McGrath, 1990). The concept of entrainment occurs when the group continues this pace in a subsequent trial, even when the time constraint has been relaxed (Kelly, Futoran and McGrath, 1990). Kelly, Futoran and McGrath (1990) also noted that groups operating under classic brainstorming instructions would most likely experience capacity problems as they attempt to generate as many ideas as possible.

Kelly, Futoran and McGrath (1990) found that on the initial trial, groups work at a faster (production) rate with the shorter initial time limit and the higher task load. The study also found that groups working on the unusual-uses tasks and beginning with the short time period will generate ideas at about the same rate by the end of the time period as in the beginning. It is thought that these groups will only experience a capacity problem and they leave the time period believing that they will simply complete as many ideas as the time period permits. If the group has problems of the same general difficulty and at least the same problem load in the next longer time period, the group is likely to work at nearly the same rate during the longer time period to solve the capacity problem experienced by the group during the first time period (Kelly, Futoran and McGrath, 1990).

Kelly, Futoran and McGrath (1990) also provided a post-test questionnaire to group members. The questionnaire asked participants "how much they had felt stress" during the time period (p. 308). The word "stress" was not defined. The questionnaire also asked the participants to note the cause of their stress. Over two-thirds of the group members who answered that question noted their stress was caused by time pressure. This study also noted a negative relationship between rate of idea generation and quality of performance (Kelly, Futoran and McGrath, 1990).

Kelly and Karau (1993) have completed the most recent study investigating time pressure and entrainment. Their study was designed to investigate both the "initial and persisting effects of time on group creativity" (p. 179). This study also utilized the unusual-uses tasks that have traditionally been used to measure group creativity when utilizing brainstorming rules (Lamm and Trommsdorff, 1973; Diehl and Stroebe, 1987).

Kelly and Karau (1993) used 99 undergraduate students to participate in the experiments. The students were placed into triads to complete the group work. The experimental design was a 3 (trials) X 3 (time limit order) factorial. The three orders of time limits were 4, 8, and 12 minutes; 8, 8, and 8 minutes; or 12, 8, and 4 minutes. This study also utilized common objects as the materials for the research. These materials were taken from Kelly, Futoran and McGrath (1990).

Kelly and Karau (1993) used both quantity and creativity-dependent measures. They calculated the rate of use as the number of uses generated by the group per minute. Scale values were determined for both creativity and feasibility of ideas. These scale values were determined through a seven pile sort-resort procedure as identified by Hackman, Jones and McGrath (1967). Inter-rater reliability was also calculated as the correlation between the judges' ratings (Kelly and Karau, 1993).

As a part of this research, two judges created 14 hierarchical categories that sufficiently categorized each of the uses generated by the groups. This categorization is similar to that of Vroom, Grant and Cotton (1969) and Lamm and Trommsdorff (1973). A variety of calculations could then be completed including the mean number of categories produced, number of categories generated per minute, and number of uses generated per category (Kelly and Karau, 1993).

During the experiment, two three-person, same-sex groups were utilized. Participants were randomly assigned to both seat positions and groups. A facilitator instructed the group, giving the following information: (a) the purpose of the study was to examine creative idea generation in three-person groups, (b) the groups were asked to generate creative and unusual uses for the common objects, and (c) the groups were instructed to focus on both quantity, originality, and feasibility of the uses. The groups were then given common objects and asked to produce 15 uses for each object. Just prior to the trial the groups were given the time constraint. At the conclusion of each trial, the groups were asked to complete a questionnaire containing questions on both performance and members' perceptions

of each trial (Kelly and Karau, 1993).

This study found that on the first trial, rate and creativity may be inversely related. The study also found that faster rates led to lower creativity while slower rates led to higher creativity (Kelly and Karau, 1993). Across-trial ratings, however, show that creativity increased over the three trials in the 4, 8, and 12 minute session and in the 12, 8, and 4 minute session. For the decreasing time series, there is a gain over trials in creativity and rate. Therefore, time pressure does not necessarily lead to reduced creativity in group performance (Kelly and Karau, 1993).

3.2 Hypotheses

We are interested in determining the kinds of treatments that yield the largest rates of idea generation with the highest level of creativity. The prior research on social entrainment and group work leads us to the following hypotheses:

> H1: The mean rate of idea generation is unequal in groups operating under differing time-constraints.
>
> H2: The mean rate of the creativity of ideas is unequal in groups operating under differing time constraints.

The key to conducting research to test these hypotheses empirically involves using appropriate tasks (complexity and motivation), appropriate timing (related to task complexity), appropriate subjects (students), and an appropriate environment (a decision support room with excellent collaborative computing software).

4 Methods

4.1 Subjects

Subjects were undergraduate students enrolled in general business courses at a private southern college. One hundred and two students participated in 34 3-member, same-gender groups. Gender was evenly distributed over conditions. Previous research involving students as subjects has been validated by a number of researchers. Students provide ready subjects, receive course credit for taking part in experiments (providing motivation), are generally given familiar enough tasks to make it possible for them to perform, and provide a ready control group which cannot be provided in field studies. Many GSS studies have used students (e.g., see Dennis and Gallupe, 1993). There has been much debate in the literature over whether students make appropriate subjects for behavioral research; notably, McGrath (1982), Gordon, Slade and Schmitt (1986), Greenberg, Gordon, Slade and Schmitt (1987), Dobbins, Lane and Steiner (1988), and Slade, Gordon, Dobbins, Lane and Steiner (1988). In summary, researchers have established that if the task is appropriate to the students, they behave reasonably, else there may be problems, i.e., the researcher must fit the task to the students' understanding, ability and motivation.

Group size should be optimized to the task (Dennis and Gallupe, 1993; Gallupe, Dennis, Cooper, Valacich, Bastianutti and Nunamaker, 1992). For our study, there was an upper limit imposed by the room configuration. Brainstorming with a group size of 1 is not a group study. The length of time of task is somewhat related to the group size in electronic brainstorming. In our pilot tests, we observed that ideas repeat in the later studies. Since there is a relatively finite set of unusual uses that is generated, we decided that groups of size 3 would be appropriate in conjunction with the time limits imposed. Larger groups would generate more ideas quickly due to the parallelism of the GSS, indicating that even shorter session times would be necessary to impose time pressure, which would pose measurement problems. Likewise, longer times would require smaller groups, which would be difficult. Finally, some of the earlier validated studies used groups of size three. The use of same sex groups was adopted because of previous research. Perhaps certain distractions can be held to a minimum this way. Again, this is another control imposed on the experimentation.

4.2 Tasks

The tasks consisted of three different unusual-uses tasks utilizing three common objects as noted by Kelly and Karau (1993). The unusual-uses tasks have been validated (Dennis and Gallupe, 1993) for GSS studies. They provide relatively simple topics that can be used for brainstorming by almost anyone, without any specialized knowledge, such as that of an expert who is highly motivated to solve a given problem. The unusual-uses tasks may even provide motivation in that they are understandable and doable, whereas attempting to solve a real-world problem such as high-crime or parking in a creative way would, no doubt, require the specialized expertise.

Three objects as identified by Kelly and Karau (1993) were used to generate ideas within the groups. These objects included the following: (a) coffee cup, (b) shoelace, and (c) paper clip. These objects were counterbalanced across conditions and trials as noted by Kelly and Karau (1993).

Subjects were instructed to generate "as many creative and unusual uses as possible" for each of the objects in three separate time trials. Each 3-member group participated in three time trials utilizing a different object for each trial. The order of the objects was random for the trials. Every 30 seconds, the group facilitator verbally noted the time remaining in the exercise.

Each group performed three idea generating tasks under three different treatments of sets of time periods. Within the treatments, the time period lengths were designated as follows: one time period was considered to be high pressure (3 minutes), the second period provided adequate time to perform the task (6 minutes), and the third time period

provided more time than necessary to perform the task (9 minutes). Task appropriateness in our context is based on task complexity, group size, and the time allocated to complete the task. All three factors are related. As mentioned above, the GSS process gain of parallelism impacts directly on all three factors. Parallelism in GSS required us to reduce the task times that were used in prior studies (e.g., Kelly and Karau, 1993). In pilot testing, small groups of size 3 could complete the task adequately in six minutes. Nine minutes provided an excess of time, while three minutes was inadequate.

Based on prior research, we expect decreases in performance in the 3, 6, 9 minute treatment as we move from 3 to 6 to 9 minutes; moderate increases in performance in the 6, 6, 6 minute treatment; and the highest increase in performance in the 9, 6, 3 minute treatment.

4.3 Independent Variables

The independent variable studied was the set (treatments) of time periods allotted during the experiment. Three sets of time periods were used. These sets of time periods required each group to work for one set of the three sets of time periods. The sets of time periods were 3, 6, and 9 minutes; or 6, 6, and 6 minutes; or 9, 6, and 3 minutes. The groups were randomly assigned to one of the three cases. Participants were also instructed to follow Osborn's (1957) brainstorming rules.

4.4 Software and the Decision Support Room

The GSS computer software used for this study is GroupSystems for Windows. This software has a strong reputation in the marketplace as being one of the best GSS packages. GroupSystems for Windows has most features needed for running effective electronic meetings in a same time / same place GSS setting. GroupSystems presents a horizontally split screen to the user. The ideas input by all participants are displayed on the upper part of the screen. A separate window in the lower part of the screen allows the user to type in his/her ideas and submit them into the common area. The software can be controlled and monitored from the facilitator workstation located in the front of the Decision Support Room at the School of Business and Entrepreneurship at Nova Southeastern University, Fort Lauderdale, FL, where the experiments were conducted. As part of the experiment, an electronic brainstorming training task was conducted to familiarize the subjects with the system.

4.5 Measures

Both quantity and creativity-dependent measures were used. The rate of idea generation was calculated as the number of non-redundant ideas generated by the group in 30 second intervals. Creativity scale values for each generated idea were determined through a sort procedure described by Diehl and Stroebe (1987) and Gallupe et al. (1992). In this procedure, two raters independently sorted all the relevant[2], nonredundant[3] ideas generated by the groups based upon a five-point scale. The scale ranged from 1 (very low creativity) to 5 (very high creativity). Raters were given definitions for each of the anchor points on the scale.

Following the work of Hackman, Jones and McGrath (1967, . 389), an idea was rated as a 1 (very low creativity) to the extent that it is "ordinary, everyday, or usual in content". An idea was rated 5 (very high creativity) if it is "unique, fresh, unusual, surprising, or refreshing". The two raters were defined as in agreement if their ratings were within one point of each other. Reliability was then calculated as the correlation between the ratings assigned by the two independent raters (Diehl and Stroebe, 1987; Gallupe et al., 1992; and Dennis, Valacich, Carte, Garfield, Haley and Aronson, 1997). Inter-rater reliability was adequate (r=.98). It would have been possible to allow the subjects to rate the ideas, however, this would have introduced another undesirable degree of freedom into the experiment.

5 Results

5.1 Idea Generation Rate

The mean and standard deviations for the measure of the idea generation rate, in 30 second intervals, of the 3, 6, 9 minute group are shown in Table 1. The results of the statistical analyses are depicted in Table 2. ANOVA analyses of groups utilizing this 3, 6, 9 minute treatment resulted in F(2,429)=0.82, p=0.442.

Table 1: Group Overall Means for Idea Rate (3, 6, 9 Group)

Group	N	Mean	StDev
3 minute	72	3.78	1.83
6 minute	144	3.63	1.57
9 minute	216	3.55	1.58

The mean and standard deviations for the measure of the idea generation rate, in 30 second intervals, of the 6, 6, 6 group are shown in Table 3. The results of the statistical analyses are depicted in Table 4. ANOVA analyses

[2]Relevant ideas were used in our study. Sometimes subjects will enter irrelevant (non-task related) ideas such as "I'm getting hungry!" or "How long are we supposed to do this?" These are omitted from the study as distracters.

[3]Nonredundant ideas were chosen because in terms of productivity, two redundant ideas simply count as one contribution toward the brainstorming effort. However, it is possible to have closely related, nonredundant ideas (i.e., for the tea bag problem: "Fishing nets for boat or dock fishing" and "fishing nets for home acquariums.") or ideas that develop as a result of other ideas (again, for the tea bag problem: "mosquito netting for camping", and "sun shading for camping."). In both cases, these would be counted as two ideas.

Table 2: ANOVA for Idea Rate (3, 6, 9 Group)

Source	DF	SS	MS	F	P
Factor	2	4.29	2.14	0.82	0.44
Error	429	1126.19	2.63		
Total	431	1130.48			

groups utilizing this 6, 6, 6 minute treatment resulted in F(2,357)=5.33, p=0.005.

Table 3: Group Overall Means for Idea Rate (6, 6, 6 Group)

Group	N	Mean	StDev
6 minutes	120	3.66	1.49
6 minutes	120	4.23	1.78
6 minutes	120	4.38	2.11

Table 4: ANOVA for Idea Rate (6, 6, 6 Group)

Source	DF	SS	MS	F	P
Factor	2	34.87	17.44	5.33	0.01
Error	357	1168.28	3.27		
Total	359	1203.16			

The mean and standard deviations for the measure of the idea generation rate, in 30 second intervals, of the 9, 6, 3 group are shown in Table 5. The results of the statistical analyses are depicted in Table 6. ANOVA analyses of groups utilizing this 9, 6, 3 minute treatment resulted in F(2,429)=38.94, p=0.000.

Therefore, the ANOVA analyses for the three time set treatments support H1. For all groups, a decrease in the mean rate occurs during the initial periods of the testing. Additionally, all three groups experienced an increase in the mean rate of ideas during the last 30 second interval. It is important to note that the overall mean rate of idea generation for the 3, 6, 9 minute treatment decreases as the amount of time increases, while those of the 6, 6, 6 minute and 9, 6, 3 minute treatments both increase, with the most dramatic increase in the latter case, for which we observe the entrainment effect. Finally, though we did not measure the levels of boredom or satisfaction, we observed that there was a sharper tapering off of the typing activity towards the end of the 9 minute time periods, and more so for the 3, 6, 9 minute treatments, than for the 9, 6, 3 minute treatments.

5.2 Idea Creativity

The mean and standard deviations for the idea creativity measures of the 3, 6, 9 group are shown in Table 7. The

Table 5: Group Overall Means for Idea Rate (9, 6, 3 Group)

Group	N	Mean	StDev
9 minutes	216	2.57	1.30
6 minutes	144	3.08	1.42
3 minutes	72	4.29	1.85

Table 6: ANOVA for Idea Rate (9, 6, 3 Group)

Source	DF	SS	MS	F	P
Factor	2	161.75	80.88	38.94	0.00
Error	429	890.97	2.08		
Total	431	1052.72			

results of the statistical analyses are depicted in Table 8. ANOVA analyses of groups in this treatment resulted in F(2,429)=0.19, p=0.827.

Table 7: Group Overall Means for Idea Creativity (3, 6, 9 Group)

Group	N	Mean	StDev
3 minutes	72	1.21	0.28
6 minutes	144	1.19	0.31
9 minutes	216	1.20	0.28

The mean and standard deviations for the idea creativity measures of the 6, 6, 6 group are shown in Table 9. The results of the statistical analyses are depicted in Table 10. ANOVA analyses of groups in this treatment resulted in F(2,357)=4.04, p=0.018.

The mean and standard deviations for the idea creativity measures of the 9, 6, 3 group are shown in Table 11. The results of the statistical analyses are depicted in Table 12. ANOVA analyses of groups in this treatment resulted in F(2,429)=0.52, p=0.594.

The ANOVA analyses for the three treatments support H2. One area of interest is the low overall mean values of the creativity of generated ideas. Prior research in a related study noted similar creativity means (Kelly and Karau, 1993). Based upon the mean values for creativity generated by this study, there is support for the claim by Kelly and Karau (1993) that groups under more time pressure generate a faster mean rate of ideas but generate ideas of lower creativity than groups under less time pressure.

6 Discussion

The results of the experiment suggest that time pressure does directly impact the rate of ideas generated by groups performing electronic brainstorming in a GSS setting. The

Table 8: ANOVA for Idea Creativity (3, 6, 9 Group)

Source	DF	SS	MS	F	P
Factor	2	0.03	0.02	0.19	0.83
Error	429	37.12	0.09		
Total	431	37.15			

Table 9: Group Overall Means for Idea Creativity (6, 6, 6 Group)

Group	N	Mean	StDev
6 minutes	120	1.29	0.32
6 minutes	120	1.20	0.24
6 minutes	120	1.21	0.21

ANOVA statistical testing supports the concept of social entrainment described by Kelly and Karau (1993). Therefore, initially placing groups of individuals into high pressure brainstorming situations seems to result in sustained high pressure performance even when these individuals are later given extended time to complete similar tasks (see Tables 1 and 2). Conversely, when individuals are given longer initial periods to complete a task, these individuals increase their productivity as the time pressure is increased (see Tables 5 and 6). Therefore, it follows that managers must carefully consider the time management and time allotment that employees are provided to complete tasks. Special conditions in collaborative computing environments (like synchronicity and even asynchronicity) require knowledge about how to impose or relax time pressure constraints on a group. Starting employees under high pressure conditions may create continued high pressure performance; providing too much time may be detrimental. One of the interesting findings of this study centers on idea creativity. Tables 7, 9, and 11 show surprisingly low mean scores for creativity relative to other studies on creativity (e.g., Dennis, Valacich, Carte, Garfield, Haley and Aronson, 1997). Our results indicate that individuals under any time pressure condition generate ideas that are relatively low in creativity. A critical impact here is that managers must recognize that placing individuals in high time pressure situations may result in ideas of relatively low creativity.

Table 10: ANOVA for Idea Creativity (6, 6, 6 Group)

Source	DF	SS	MS	F	P
Factor	2	0.55	0.28	4.04	0.02
Error	357	24.45	0.07		
Total	359	25.00			

Table 11: Group Overall Means for Idea Creativity (9, 6, 3 Group)

Group	N	Mean	StDev
9 minutes	216	1.17	0.34
6 minutes	144	1.20	0.31
3 minutes	72	1.17	0.18

Table 12: ANOVA for Idea Creativity (9, 6, 3 Group)

Source	DF	SS	MS	F	P
Factor	2	0.10	0.05	0.52	0.59
Error	429	40.13	0.09		
Total	431	40.23			

7 Limitations and Future Research

There are several limitations associated with this study. First, undergraduate business students were utilized in this study. Utilizing students in research that generates ideas may limit those ideas based upon the participant's limited experiences. Additionally, the students were given a portion of their final course grade for simply participating in the research. Therefore, the motivation, and as a consequence, the pressure felt by the students may not have been as great as would occur in a business environment. However, using students as subjects in experimental treatments comparing different approaches is generally considered a valid research approach as are the results (see McGrath, 1982; Gordon, Slade and Schmitt, 1986; Greenberg, Gordon, Slade and Schmitt, 1987; Dobbins, Lane and Steiner, 1988; and Slade, Gordon, Dobbins, Lane and Steiner, 1988). Field studies in real business environments are impossible to control or repeat. Though, to better understand the impact of time pressure, it would be appropriate to further study this variable utilizing individuals from the business environment in field studies. Further research is appropriate for both non-management as well as management groups. Studying the differences and/or similarities between these two groups may also provide additional insights into time pressure studies.

Second, the tasks used in this study, though attempting to inspire creativity, were simplistic in nature. This of course helps when using student subjects, so long as sufficient time is allocated for the given level of task complexity. Although the tasks were taken from prior GSS research and therefore grounded in the literature, care should always be taken when generalizing the results across all idea generating tasks. Investigating differing research tasks is an area in which further research is suggested.

Third, even though two individuals rated idea quality (creativity) with high inter-rater reliability, it is possible that the creativity scale was biased or inappropriate, or that

the ideas generated may not really have been very creative. Many low-rated creative ideas may weigh more heavily than a few, very creative ideas. In a real-world decision making situation, a few high-quality ideas may be the optimal solution space to the problem being solved, but, the creativity measure allows for many low-rated creative ideas to indicate a better job of brainstorming. This is a weakness of the creativity measure. In addition, in a real-world decision making situation, the participants generally discard duplicate, irrelevant and low-quality ideas. This, however, would have skewed the study by adding another degree of freedom.

Fourth, the facilitator announced the time remaining for each exercise every 30 seconds. During the final 30 second intervals, participants increased the mean rates of their tasks which does occur in real-world brainstorming situations. Additional research in which the participants are not informed about the remaining time and the treatment abruptly ends without warning would be beneficial to determine if this trend remains constant.

Fifth, 3 minute blocks of time may have been too long (or too short) for the subjects to perform their task. Perhaps treatments consisting of shorter time intervals such as 1.5, 3 and 4.5 minutes, or 1, 2, and 3 minutes, would have been more suitable.

Sixth, learning curve influences may have occurred in that the students became more proficient with the software or the brainstorming task itself, so that early efforts may have not been performed at the peak level of performance.

8 Implications

This work, and potential extensions, have some interesting impacts on how managers can incorporate collaborative computing technologies, and on how GSS could be designed and utilized in practice. This includes newer collaborative systems that do not require all participants in a meeting to work in the same place at the same time. Time pressure exists even for groups working asynchronously.

Work groups are affected by time pressures, independent of special technology. A good manager should have an expectation of how much time an employee, or a work group, requires to perform a task at sufficient levels of competency and excellence. Now, managers must develop an awareness of how collaborative computing technologies, such as GSS, impact on task performance. A good manager must adjust his/her expectations for the group, and allow an appropriate amount of time to perform tasks at hand, especially when using GSS. A manager can then impose varying time pressures for each task at hand to create a work environment to produce desired results, e.g., the quality level and amount of work performed. A good manager should know that too much time can be just as detrimental as not enough time for task completion.

GSS can be designed to incorporate time pressure effects induced by a facilitator by indicating how close the group is to using their time allotment in completing various tasks. In synchronous mode, i.e., groups working in a "decision" room – the same time / same place situation, induced time pressure could keep the group 'synchronized' and move them along throughout their tasks while using GSS. Time pressure effects could be more dramatic in asynchronous, dispersed groups, i.e., groups using Web-based groupware (see Turban and Aronson 1998) - the different time / different place situation. In this latter case, time pressure is automatically imposed by creating strict deadlines (synchronization points) so that the entire group will know when it can view their results and thus know when to move on to the next phase of the meeting to make further contributions. This is critical for dispersed groups in multiple time zones, which is not uncommon in distance learning environments. The multiple time zone situation causes other problems, and managers must take care to include all members in the process. In either case, time pressure and the effects of entrainment could generally be imposed by the facilitator in consultation with the manager of the group when setting the agenda for the meeting.

Further work can include using different sets of times, tasks other than brainstorming (such as decision making tasks), multiple-step electronic group meetings, asynchronous group environments, real-world environments, comparison of GSS time pressure effects with those of groups in manual mode, the relationship of time pressure to member satisfaction, and others.

References

[1] Dennis, A.R. and R.B. Gallupe (1993), A history of GSS empirical research, Chapter 3 of Jessup, L.M. and J.S. Valacich, *Group support systems: New perspectives, Macmillan Publishing Company*, New York, 59-77.

[2] Dennis, A.R., J.F. George, L.M. Jessup, J.F. Nunamaker and D.R. Vogel (1988), Information technology to support electronic meetings, *MIS Quarterly*, 12, 4, 591-624, December.

[3] Dennis, A.R., J.S. Valacich, T. Carte, M. Garfield, B. Haley J.E. and Aronson (1997), The effectiveness of multiple dialogues in electronic brainstorming, *Information Systems Research*, 8, 1-9.

[4] DeSanctis, G. and R. Gallupe (1987), A foundation for the study of group decision support systems, *Management Science*, 33, 5, 589-609.

[5] Diehl, M. and W. Stroebe (1987). Productivity loss in brainstorming groups: Toward the solution of a riddle, *Journal of Personality and Social Psychology*, 53, 497-509.

[6] Dobbins, G.H., I.M. Lane and D.D. Steiner (1988), A note on the role of laboratory methodologies in applied behavioural research: Don't throw out the baby with the

bath water, *Journal of Organizational Behavior*, 9, 3, July, 281-286.

[7] Gallupe, R.B., L.M. Bastianutti and W.H. Cooper (1991), Unlocking brainstorms, *Journal of Applied Psychology*, 76(1), 137-142.

[8] Gallupe, R.B., A.R. Dennis, W.H. Cooper, J.S. Valacich, L.M. Bastianutti and J.F. Nunamaker (1992), Electronic brainstorming and group size, *Academy of Management Journal*, 35, 350-369.

[9] Gordon, M.E., L.A. Slade and N. Schmitt (1986), The 'science of the sophomore' revisited: From conjecture to empiricism, *Academy of Management Review*, 11, 1, Jan., 191-207.

[10] Greenberg, J., M.E. Gordon, L.A. Slade and N. Schmitt, (1987), The college sophomore as guinea pig: Setting the record straight/student guinea pigs: Porcine predictors and particularistic phenomena, *Academy of Management Review*, 12, 1, Jan., 157-163.

[11] Greenberg, J., K. Williams and M. O'Brien (1986), Considering the harshest verdict first: Biasing effects on mock juror verdicts, *Personality and Social Psychology Bulletin*, 12, 41-50.

[12] Hackman, J.R. (1966), Effects of task characteristics on group products, Technical report No. 5, Urbana, IL: Department of Psychology, University of Illinois.

[13] Hackman, J.R., L.E. Jones and J.E. McGrath (1967), A set of dimensions for describing the general properties of group-generated written passages, *Psychological Bulletin*, 67, 379-390.

[14] Janis, I.L. (1982), *Victims of groupthink*, Boston: Houghton Mifflin.

[15] Karau, S.J. and J.R. Kelly (1993), The effects of time scarcity and time abundance on group performance quality and interaction process, *Journal of Experimental Social Psychology*, 28, 542-571.

[16] Kelly, J.R., Futoran, G.C. and McGrath, J. E. (1990). Capacity and capability: Seven studies of entrainment of task performance rates, *Small Group Research*, 21, 283-314.

[17] Kelly, J.R. and Karau, S. (1993), Entrainment of creativity in small groups, *Small Group Research*, 24(2), 179-198.

[18] Kelly, J.R. and J.E. McGrath (1985), Effects of time limits and task types on task performance and interaction of four-person groups, *Journal of Personality and Social Psychology*, 49(2), 395-407.

[19] Lamm, H. and G. Trommsdorff (1973), Group versus individual performance on tasks requiring ideational proficiency (brainstorming), *European Journal of Social Psychology*, 3, 361-387.

[20] McGrath, J.E. (1982), Dilemmatics: The study of research choices and dilemmas, in McGrath, J.E. (ed.), *Judgment calls in research*, Beverly Hills, CA: Sage, 69-80.

[21] McGrath, J.E. (1984), *Groups: Performance and interaction*, Englewood Cliffs, NJ: Prentice Hall.

[22] Mullen, B., D. Johnson and E. Salas (1991), Productivity loss in brainstorming groups: A meta-analytic integration, *Basic and Applied Social Psychology*, 12, 3-23.

[23] Myers, R.M. (1997). The impact of time pressure on idea generation: An investigation of productivity and creativity utilizing computer supported groups, Unpublished doctoral dissertation, Fort Lauderdale, FL: School of Business and Entrepreneurship, Nova Southeastern University.

[24] Nunamaker, J.F., Jr., A.R. Dennis, J.F. George, J.S. Valacich and D.R. Vogel (1993), Group support systems research: Experience from the lab and field, Chapter 7 in Jessup, L.M. and J. Valacich (eds.), *Group support systems: New perspectives*, New York: Macmillan.

[25] Osborn, A.F. (1957), *Applied imagination*, New York: Scribner.

[26] Reinig, B.A., R.O. Briggs, S.A.Brandt and J.F. Nunamaker, Jr. (1997), The electronic classroom on fire: Why it happens, and how to put out the flames, *Proceedings of the Thirtieth Annual Hawaii International Conference on Systems Sciences*, Wailea, HI.

[27] Slade, L.A., M.E. Gordon, G.H. Dobbins, I.M. Lane and D.D. Steiner (1988), On the virtues of laboratory babies and student bath water: A reply to Dobbins, Lane, and Steiner: A further examination of student babies and laboratory bath water: A response to Slade and Gordon, *Journal of Organizational Behavior*, 9, 4, Oct., 373-378.

[28] Thurow, L. (1980), *The zero-sum society: Distribution and the possibilities for economic change.* New York: Basic Books.

[29] Turban, E. and J.E. Aronson (1998), *Decision support systems and intelligent systems*, 5th ed., Upper Saddle River, NJ: Prentice Hall.

[30] Vroom, V.H., L.D. Grant and T.W. Cotton (1969), The consequences of social interaction in group problem solving, *Organizational Behavior and Human Performance*, 4, 77-95.

[31] Wright, P. (1974), The harassed decision maker: Time pressures, distractions, and the use of evidence, *Journal of Applied Psychology*, 59(5), 555-561.

[32] Zakay, D. and S. Wooler (1984), Time pressure, training, and decision effectiveness, *Ergonomics*, 27(3), 273-284.

Facilitating and Coordinating Distributed Joint Applications Development

James Suleiman
University of Colorado, College of Business, Colorado Springs, Colorado, USA,
Tel: 719 262-3335, Fax: 719 262-3494, E-mail: jsuleima@mail.uccs.edu
AND
Roberto Evaristo
University of Denver, Daniels College of Business, Information Technology and Electronic Commerce Department, Denver, Colorado, USA,
Tel: 303 871-4340, Fax: 303 871-2016, E-mail: evaristo@du.edu
AND
Gigi G. Kelly
The College of William & Mary, School of Business Administration,
Williamsburg, Virginia, USA,
Tel: 757 221-2970, E-mail: ggkell@business.wm.edu

Keywords: Information systems, Joint applications development, facilitation, coordination, distributed work environments, group processes

Edited by: Gary Klein
Received: June 11, 1999 **Revised:** December 9, 1999 **Accepted:** December 14, 1999

Virtual teams have become a fixture of organizations in the 90s. Part of the reason for the creation of virtual teams is the existence of limited resources or need for people to share knowledge concurrently over long distances. Software development is not an exception to this problem, and it is likely to become even more of a distributed process in the near future. In this paper we address the problems that are likely to be found in distributed Joint Application Design (JAD) efforts. We then propose an interpretivistic study to perform a more complete analysis of the key issues in distributed JAD. Important applications to practice are raised.

1 Introduction

IBM developed Joint Application Design (JAD) in the late 1970's as a process for collecting information system requirements and reviewing system designs. The premise of JAD is to add structure to the requirements definition phase of analysis by involving key stakeholders in structured meetings run by a session leader (Hoffer, George et al. 1998). The JAD session leader is often asked to wear many hats (e.g., group facilitator, systems analyst, and sometimes project leader). In JAD sessions, participants share knowledge and documents pertaining to requirements definition including process models, data models, and other documents and diagrams. Computer Aided Software Engineering (CASE) tools help transform unstructured information into structured models that can be shared by all participants. The main purpose of JAD perhaps is best summarized in the following excerpt from (Hoffer, George et al. 1998):

> The primary purpose of using JAD in the analysis phase is to collect systems requirements simultaneously from the key people involved with the system. The result is an intense and structured, but highly effective, process. As with a group interview, having all the key people together in one place at one time allows analysts to see where there are areas of agreement and where there are conflicts. Meeting with all these important people for over a week of intense sessions allows you the opportunity to resolve conflicts, or at least to understand why a conflict may not be simple to resolve.

One of the newer problems faced by JAD proponents is that often it is infeasible to have groups meet at the same time and in the same place due to geographic separation, difficulty in scheduling all members, and other constraints. Although information technology (IT) easily allows dispersed or distributed work groups to communicate, the lack of face-to-face (FTF) involvement has provided new issues to address in the systems development environment (Bandow 1998). This paper presents our findings from our initial research project that investigates the coordination and facilitation of distributed groups conducting JAD. First we present a limited review of the current literature regarding distributed JAD and facilitation of JAD sessions. Next, the methodology employed in this initial project is explained. We conclude the paper with a discussion of our findings and conclusions. We have used the findings presented here as the basis for on going and future research projects related to distributed environments for group work.

2 Distributed JAD

Drawing on the literature of traditional JAD research, critical elements have been identified in order to increase the likelihood of effective JAD sessions. (Carmel, Whitaker et al. 1993) identified the following four building blocks:

1. Facilitation. A designated leader (or leaders) manages the meeting. Some JAD practitioners consider the meeting leader to be key to process success, even more so than the act of gathering the users in one place, the essence of JAD.

2. Agenda setting/structure. The meeting must have a plan of action.

3. Documentation. One or more designated scribes carefully document everything in the meeting. Various lists are rigorously maintained.

4. Group Dynamics. Group dynamics techniques are used for inspiring creativity (e.g., brainstorming), resolving disagreements (e.g., airing facts, documenting them as "issues," taking notes), and handling speaking protocols (e.g., enforcing "only one conversation at a time").

Transferring these factors into a distributed setting is not necessarily an easy task. These building blocks become more difficult to manage when groups become distributed geographically, and some of the key structures of JAD may become infeasible due to this distribution.

Before examining the possible effects of distribution on traditional JAD, it is important to illustrate how groups can be distributed. Not only can groups be distributed by time and space (Johansen 1988), but they can also be proportionately distributed by members. For example, some groups may have geographically co-located members and other geographically co-located members that are at another location. This brings to light a third category, degree of distribution. Therefore, distributed groups can vary in time (same/different), place (same/different), and degree of distribution (partial/complete). Please see figure 1 for an illustration.

With traditional JAD everyone meets at the same time often for several days at an off-site location thus providing an environment absent from other distractions. This scenario is unlikely to occur with distributed JAD due to normal business demands on participants that they are unlikely to be able to divorce themselves from. Furthermore, there is often a lack of social norms of a typical group, and there is no particular consequence for missing an action. On the other hand, research by (Evaristo, Scudder et al. 1998; Evaristo and Fenema 1999)suggests that critical to the success of distributed projects are: (1) a initial face to face meeting where all the stakeholders meet and get to know each other; (2) a traditionally scheduled meeting, for instance once a week, where everybody meets even if on a video teleconferencing link; this should happen even when there is no specific agenda to be discussed as subjects always come up anyway. It is the opinion of the authors that most distributed JAD in the future will be a combination of synchronous and asynchronous place and time with distribution of participants uncertain (e.g., can be partial or complete). It is also important to note that the facilitator can be distributed, or co-located, and that there may be multiple facilitators.

The role of the facilitator has been identified as one of the critical elements for successful traditional JAD. Exactly how this role is enacted in a distributed environment is just beginning to unfold. The rapid growth and adoption of the Internet, desktop conferencing, and telecommuting are requiring workers to participate in distributed meeting although many participants are ill prepared to effectively contribute in a distributed environment (Kelly and Bostrom 1998). When groups are distributed geographically the dynamics of the group change and this can often produce many challenges in facilitation. FTF groups exhibit more effective leadership and coordination competence over their distributed counterparts (Burke and Chidambaram 1995). The importance of facilitation is heightened in a distributed JAD environment. The facilitator is often the person appropriating the technology. To maximize the effectiveness of this new process, facilitators must be aware of the components that make-up the JAD environment. They must be skilled at helping others address the task before them. The facilitator not only facilitates the actual JAD sessions but is also responsible for many pre- and post- JAD session activities. It is often the facilitator's job to perform the agenda setting and providing structure that is identified as a critical JAD element. In addition, facilitators must also be skilled at helping group members deal with the group dynamics aspects of the changes they are attempting to make, especially when these changes are radical, as is often the case with JAD (Conner, 1992). For example, group dynamics are an important aspect in getting a group to agree on task issues.

> In order for any group to be productive, it must give attention to task accomplishment. That is, the work of different members must be coordinated and combined so that everyone is pulling together to attain the desired outcomes of the group session. At the same time, the group also must be mindful of the emotional and personal welfare needs of the members. If not, the group jeopardizes its ability to accomplish its task. Most seriously, if proper maintenance behaviors are not performed, the survival of the group is threatened. (Kayser, 1990, pp. 84-85)

Socio-emotional behaviors associated with group dynamics, on average, account for one-third of the group processes with positive socio-emotional behaviors twice as likely as negative behaviors (Ridgeway and Johnson, 1990). A need for balance between task and social behaviors is required (Chidambaram and Bostrom, 1996; Kayser,

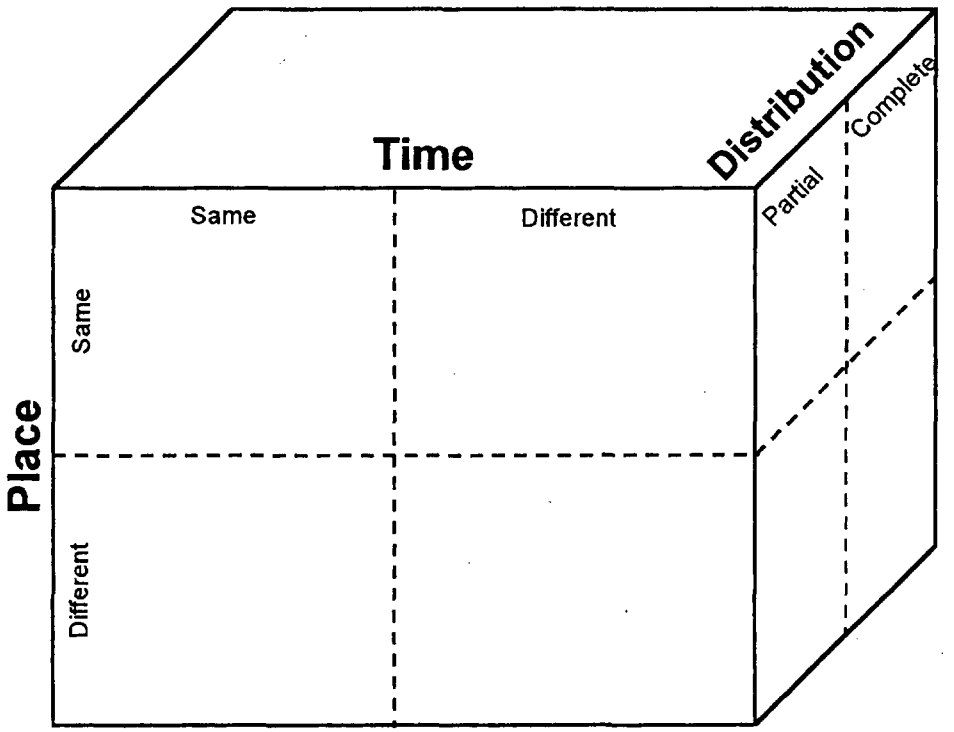


Figure 1: The Time/Place/Distribution continuum

1990; Dutton, 1987). There is no prescribed best blend; however, the group dynamics issues are often disregarded in order to accomplish the task (Ellis and Fisher, 1994). Disregarding the group dynamics can be harmful to the accomplishment of the JAD session outcomes. An overemphasis on task functions may increase a group's short-term performance; however, a continued focus on task needs alone can undermine the long-term effectiveness of the group (Bowditch and Buono, 1990).

Drawing from tradition facilitation research and literature, (Bostrom, Anson et al. 1993) identify a number of effective interventions of group processes that are used in the traditional meeting environment (see Table 1). These interventions have been identified in empirical and case studies. These interventions are important for effective JAD sessions; however, it is not clear how to appropriate these interventions in a distributed environment.

Several key points were identified when reviewing these studies (Bostrom, Anson et al. 1993). First, broader interventions that support both effective task and group dynamic processes are superior to narrowly focused interventions. This supports the argument previous made regarding the necessity of balance between task and group dynamic related outcomes. Another key finding is that the effectiveness of a facilitator is associated with training. Highly trained facilitators are more effective than facilitators are with less training. Training facilitators for distributed meetings is truly in it infancy with very limited research. Finally, meetings are more productive when participants are provided some facilitation training (Hall and Watson 1970).

Clearly, facilitators must be aware of not only the task but also the group dynamic dimension in order to best utilize the JAD participants as resources.

The small amount of distributed group research and some mixed findings make it very difficult to form any hypothesis with regards to group dynamics and group performance. Studies have shown member participation to be just as cohesive and equal in distributed groups (Burke and Chidambaram 1995), while other studies show that distributed groups have a greater tendency towards centralization (i.e., a greater differentiation between leaders and peripherals), (Haythornthwaite 1999). Research in distributed applications development has found FTF contact to be critical to project success, especially at the initial formation of the group and also during regularly scheduled meetings (Bandow 1998; Evaristo, Scudder et al. 1998). While this paper does not assume that IT can replace FTF contact, it does assume that it can play an important support role when groups are unable to meet FTF. The development process and effectiveness of distributed groups is also

Table 1: Interventions That Broadly Improve Group Processes and Outcomes

1.	**Applying Structured Procedures**
	- providing instruction to group members (Hall and Watson 1970)
	- extending problem formulation (Volkema 1983)
	- extending idea generation (Ven and Delbecq 1974)
	- delaying solution adoption (Hoffman and Maier 1959)
2.	**Encouraging Effective Task Behaviors**
	- discussing task procedures (Hackman and Kaplan 1974)
	- applying explicit criteria (Hirokawa and Gouran 1989)
	- using factual information (Hirokawa and Gouran 1989)
	- maintaining focus on task goal (Dalkey & Halmer, 1963)
3.	**Encouraging Effective Relational Behaviors**
	- encouraging broad participation and influence (Hoffman and Maier 1959)
	- managing conflict constructively (Putman 1986)
	- emphasizing consensus acceptance over majority votes (Hall and Watson 1970)
	- applying active listening techniques (Bostrom 1989)
	- discussing interpersonal processes (Hackman and Kaplan 1974)
4.	**Training**
	- training group member and/or leaders (Hall and Watson 1970)
	- training external facilitators (Maier and Maier 1957; Bostrom 1989; Hirokawa and Gouran 1989; Anson, Bostrom et al. 1995)
(Bostrom, Anson et al. 1993)	

a great unknown. There has been some work explaining that distributed groups develop differently in certain aspects (Coutu 1998) while some studies question whether or not geographical distribution even matters (Pawar and Sharifi 1997).

Due to the distribution of group members in a distributed JAD session, this can have a profound effect on how to effectively manage the four building blocks mentioned previously. This leads to the two main questions that this study attempts to create propositions from:

1. How can we effectively facilitate distributed JAD?
2. How can we effectively coordinate distributed JAD?

It is important to recognize that facilitation and coordination are not mutually exclusive and are somewhat ambiguous concepts. However the facilitation question hopes to raise propositions on the role of the facilitator, including the possibility of using multiple facilitators, and what facilitation/ group dynamics techniques and tools are effective or problematic and under which conditions. Coordination will raise propositions on effective or problematic scheduling techniques, IT utilized, and providing documentation, or group memory.

In this research paper we do not try to determine whether distributed teams are "better" than traditional teams. It is the bias of the authors that, in fact, they are not but they are a necessity. As was stated by (Melymuka 1997):

> Virtual teams are counterintuitive, difficult to design, costly and complex to implement, messy to manage and far less productive than real teams. But they address the needs of a new work environment in which downsizing has left a veneer of expertise to cover a global operation or in which mergers have created a patchwork of mismatched skills and needs.

3 Methodology

A research project was created using two separate groups of students from universities A and B, about 60 miles apart. The premise of the research project was to have the two distributed groups of students participate in a distributed application analysis project for a university registration system. Each group of students was given specific roles and outcomes to achieve within a 3-week period. Students are clearly knowledge users with their own on-line course registration system and have been used in other studies as business user substitutes with a very similar task (Hickey, Dean et al. 1999) The research project as it unfolded is described below.

Initially, during class one of the instructors had the students created a preliminary use-case scenario description of the registration system of University A, and placed it on the Web. He informed his students that they would role-play "users" of the registration system and that questions about the uses and details of the registration would be forthcoming shortly. There were 52 users and all users had equal responsibility for the project. Users were instructed to answer a minimum of two questions they received from the designers.

The instructor of the students at University B informed his students that they would role-play "designers" and were required to prepare a fairly complete analysis of the registration system at University A by asking questions to the "users". The designers were separated into 6 groups of 5 students each. At the end of the 3 weeks, each group of designers had to turn in a detailed use-case scenario description of the registration system as well as individual statements that included reflections about the project and distributed communications.

An e-mail distribution list was created, and all students from both classes were given instructions on how to join. Roughly one week after the project was unveiled for both sets of students, all had subscribed to the list.

The groups of "designers" were given three weeks to complete the task of interviewing the "users" over e-mail and to deliver the required specifications documentation of the current registration system at University A. No formal facilitation or other structure were provided; however, all the students were enrolled in a Systems Analysis and Design class and had received instruction about the responsibilities of an Analyst. The students had learned about different techniques for acquiring design specifications. For the first week, there was no traffic in the distribution list. At this point, the instructors intervened and reminded the students that the project was due in two weeks.

The last two weeks saw a flurry of communication from both sides. Because no structure had been added to the distribution of emails, both users and designers became confuse and frustrated with the project. For example, some designers asked very broad questions such as "list all the inputs and outputs to the registration system." The users did not want to take the time to list all the inputs and outputs and either did not answer the question or only provided partial details. This response is not unusual in a traditional systems development project when a question of this broad magnitude is presented to users. Also, users only were told they had to answer a minimum of two questions. Approximately 72% of the questions posed were answered.

At the end of the 3-week period, each of the group of the designers turned in a use-case scenario description. The results ranged from above average to below average. Also, the designers and users turned in individual papers about the project and their thoughts and conclusions about distributed communication. The lessons learned from this initial research project have been extremely helpful in setting up a subsequent distributed research project across three universities.

The findings that will be presented in the next section are the combined results of qualitative observation of the distribution list communications plus extended debriefing with the "designers" (i.e., students of the University B). Appendix A includes the actual half-page description of the registration system of University A. It is very brief and was intended only as a starter for the project.

The interpretivistic analysis of the notes, emails, CASE and written materials resulting from this project generate the key categories of the problems or issues associated with distributed JAD. The grounded theory approach as presented in (Strauss and Corbin 1990) guides this research effort. This research is considered interpretive in that there are no formal propositions (although that will be an outcome of the preliminary study), and that our knowledge of reality will be obtained from a rich analysis of the initial studies (Klein and Myers 1999). The resulting list of issues is classified and this classification scheme serves as the foundation for future research studies related to distributed JAD.

The use of students should not be detrimental to this particular research project in that this is a preliminary investigation that will be extended to corporate groups doing distributed JAD. Furthermore, the students should benefit from learning to work in a dispersed environment.

> Business educators must help all students become adept at distanced interaction, because skills in information gathering from remote sources and in collaboration with dispersed team members are as central to the future American workplace as learning to perform structured tasks quickly was to the industrial revolution.(Dede 1996)

It is important to note that we acknowledge that, as researchers, we may have certain biases (some of which we are aware of). This can lead to socially created distortions and relates to the principle of suspicion (Klein and Myers 1999). We would like to identify some of the biases we are aware of:

- A belief that while IT will not replace FTF communications, it can act as a partial substitute.
- A belief that technology is a tool and not a means to an end.
- A belief that groups undergo an initial stage where many of the dynamics of the group and interpersonal relationships are formed.

While there is some research on distributed groups, a large portion of it deals with electronic meeting systems and does not address the management and coordination of a complex project. We feel there is a strong need to develop a richer understanding of distributed JAD in that it is becoming more commonplace due to a combination of the constraints listed previously and the increased richness of IT in communicating shared knowledge between people.

4 Findings

Several sets of findings were gleaned from the data obtained. Perhaps the most important was that no rapport was created between users and designers, partly because of the relatively short time available to perform the assignment. Suggestions on how to improve the situation will be offered

at the end of this section. The second finding addresses the task at hand. The lack of structure for asking questions and responding to emails led to frustration on both users and designers of the project. This was clearly evident in analysis of the actual content and evolution of the e-mails across the distribution list. Finally, the process that developed in this exchange of e-mails was also enlightening, seeming to suggest that the existence of a project manager or facilitator to the whole process would have been very helpful. Each of these findings is discussed in further detail below.

The designers due to confusion about what exactly they were supposed to do did basically not use the first week. However, in the beginning of the second week, there was the sudden realization of the short time ahead to complete the assignment. At that point, there was a lot of communication from the designers – several messages hit the distribution list asking a large number of fairly vague questions. The users not only largely ignored this first set of messages because they were not aware of the actual answers but also because trying to answer them would take an inordinate amount of time. At no point did the designers and the users take the time to make introductions and create a sense of groupness between the two distributed groups. The traditional "stages of small group development" developed by (Tuckman 1965) identifies five steps that groups go through: forming, storming, norming, performing, and adjourning. In this project, the designers, facing time pressure, went directly to the task at hand, largely ignoring forming and straight into storming. The building and maintaining of the relationship between the designers and users was bypassed to get the task done. Team and group development literature clearly identifies the need to for groups to build relationships in order to succeed at the task (Tjosvold and Tjovold 1995). Furthermore, the members of the team gain more personal satisfaction if the time is taken to form some sort of relationship (Musgrave and Anniss 1996). When relationships in groups evolve, members make their assessments of others on the basis of past history. Members should have information on other members' backgrounds, beliefs, and interest in establishing a good relationship. Team exercises that focus on increasing information exchange among team members and encourage commitment early might help in the groups development (Jarvenpaa, Knoll et al. 1998). First finding: addressing the task prior to establishing some form of relationships among group members can make completing the task difficult and lead to frustration among the group members. In a distributed environment, the forming of relationships appears to be a more complex issue than in traditional FTF relationships. For example, having all-inclusive meetings over chat lines may allow people to get a better picture of the scope of the project as well as set the tone for the ensuing work. This may be in fact one of the best rapport creating mechanisms to people who have never met. In future studies, we plan to investigate the building of relationships in distributed environments and utilize different rapport building techniques.

After a regroup meeting on the part of the designers during one of the regularly scheduled class, it became obvious that the absence of structure and rule from the designer group over the user group stopped them from "requesting and receiving" answers. Although it is often the case of reluctant users when analysts are trying to elicit requirements in traditional JAD, this was exacerbated in a distributed project where people did not even know each other or had an appreciation of the real level of knowledge the other side had. When designer tried gathering data, the randomness of the questions and reply to questions coupled with the lack of documentation standards led to sub-par performance of the designers. Etiquette about exchanging information needed to be created. The designers did recognize the problem and began to ask questions that were simple as well as worded in a very nice way. Second finding: structure and rules have to be clearly stated and accepted by all group members when conducting distributed JAD sessions. In this case, it would have been wise for the designers to suggested some ground rules and examples of documentation for the users to have a better appreciation of the type of material that would have aided the designers. A document repository would have been a great help.

Eventually answers started to pour in; however, several new problems surfaced. For instance, duplicate questions. And duplicate answers, because sometimes the responses instead of being addressed to the list were sent to the individual who had asked them, who by its turn would not post it to the list. Simple issues such as no multiple questions in one e-mail (a few are always ignored); how to deal with conflicting answers to the same question; very limited use of totally open-ended questions; and perhaps the most important of all, the need for real-time communication such as a chat line, perhaps with representatives of both groups are all process issues that needed to be addressed. This process learning is part of the third finding: the existence of a project manager or facilitator to the whole process would have been very helpful.

We conclude are discussion with the role of a potential facilitation mechanism. By default, we had a situation with virtually no facilitation. No rules or guidelines to communication were sent over the distribution list. We decided to call it level "0" facilitation. A logical upper level of facilitation would have been to suggest several rules for communication and make them available both for users and designers – level "1". Active involvement of an assigned facilitator would then be level "2". An active facilitator would read all the messages, make suggestions on the air and gently prod both sides to correct the process. It would also help create the rapport and trust at the beginning of the process through an all-together meeting over NetMeeting or any chat mechanism. Finally, we envisioned also a level "3" facilitation, where the actual correspondence would be mediated by the facilitator, working almost as a censor. Based on what we saw, this level seems to be too intrusive, and we would suggest level 2 as the most efficient in this case. Level 2 would also have the advantage of

involving users to a higher extent, something that has been shown to improve the quality of the system.

We have taken the view that structure can be used as a facilitation tool for distributed groups. We have avoided media choices and media richness as described by (Daft and Lengel 1986). And conversely feel that arguments against media richness (Ngwenyama and Lee 1997) and for media synchronicity (Dennis and Valicich 1999) are not applicable for this paper. The scope of this research project was exploratory and too broad to incorporate any of the media theories. It is our bias that media choice is a function of specific circumstances including task, group, history, and facilitator preference. For distributed groups media choice is a very important issue for future studies and we plan on examining this as an extension of this study.

Distributed JAD is likely to be more common in the future, particularly because of cost of traveling and committing a large number of people for a relatively extended time, limited resource and knowledge availability, and need to share knowledge on a worldwide basis over long distances. We are already seeing outsourcing across countries, but typically the almost finished set of specifications is sent to the actual developer (Kumar and Willcocks 1996). This works particularly well with strongly structured applications, such as payroll and other transaction processing systems. On the other hand, the more communication is needed across the different stakeholders in the different stages of systems development; the least likely is that development will be outsourced because of the natural impediments caused by the distance. Naturally, if the communication and coordination problems that arise in situations of highly unstructured systems could be ameliorated, then it may be claimed to be one step closer to the elusive goal of true distributed software development. Therefore, an understanding of these communication and coordination issues – including during the analysis phase over distributed environments – is fundamental to improving the long-term performance of software development productivity and effectiveness.

Finally, the need for effective facilitation in today's business environment is clearly stated by (Kayser 1990) in his book *Mining Group Gold*:

> In today's competitive, turbulent, complex, interdependent economic environment, teams, teamwork, and collaboration, by necessity, are the heart and soul of world-class organizations for the 90s and beyond. Teams have two assets that exceed those of any individual on the team: they possess more knowledge, and they can think in a greater variety of ways. The potential assets may not always be reached however. Because of poor facilitation and leadership, teams may fall into so much dysfunctional conflict that they cannot operate. On the other hand, excellent facilitation and leadership can move the team to realize its full potential and produce a superior output which propels everyone's commitment and feelings of satisfaction to their zenith. Group sessions are not to be trifled with. Your organization's success, and your own personal growth, development, and promotability, depend on your ability to seize every meeting opportunity as a forum for role-modeling facilitation and collaborative leadership. (pp. 31-32)

How to effectively facilitate meetings and JAD sessions in distributed environments is a question that demands answers – as this form of group communication becomes routine in both universities and businesses. This research project makes an initial attempt to begin to understand the complex dynamics that exist with distributed JAD sessions. These learnings provide us with some of the building blocks for future research. The distributed environment that business is embracing provides researchers with a rich and very important area to study as we enter the next century.

References

[1] Anson, R. G., R. P. Bostrom, et al. (1995). An Experiment Assessing Group Support System and Facilitator Effects on Meeting Outcomes. *Management Science*, 41, 2, p. 189-208.

[2] Bandow, D. (1998). Working with the Borg: Trust, system development and dispersed work groups. Conference on Computer Personnel Research.

[3] Bostrom, R. P. (1989). Successful Application of Communication Techniques to Improve the System Development Process. *Information and Management*, 16, p. 279-295.

[4] Bostrom, R. P., R. G. Anson, et al. (1993). *Group Facilitation and Group Support Systems. Group Support Systems: New Perspectives*. L. M. Jessup and J. S. Valacich (Eds.), New York, NY, Macmillan.

[5] Burke, K. & L. Chidambaram (1995). Developmental Differences Between Distributed and Face-To-Face Groups in Electronically Supported Meeting Environments: An Exploratory Investigation. *Group Decision & Negotiation*, 4, 3, p. 213-233.

[6] Carmel, E., R. D. Whitaker, et al. (1993). PD and joint application design: A transatlantic comparison. *Communications of the ACM*, 36, 4, p. 40-48.

[7] Coutu, D. L. (1998). Organization: Trust in Virtual Teams. *Harvard Business Review*, 76, 3, p. 20-21.

[8] Daft, R. L. & R. H. Lengel (1986). Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, 32, 5, p. 554-571.

[9] Dede, C. (1996). Emerging Technologies in Distance Education for Business. *Journal of Education for Business* 04/01/1996, p. 197-210.

[10] Dennis, A. R. & J. S. Valicich (1999). Rethinking Media Richness: Towards a Theory of Media Synchronicity. Hawaii International Conference on System Sciences, Maui, Hawaii.

[11] Evaristo, R. & P. v. Fenema (1999). A Typology for Project Management: Emergence and Evolution of New Forms. *International Journal of Project Management* forthcoming.

[12] Evaristo, R., R. Scudder, et al. (1998). Software Development in a Distributed Environment or How Virtual Teams Were Programmed to Succeed, University of Denver Working Paper.

[13] Hackman, J. R. & R. Kaplan (1974). Interventions into Group Process: An Approach to Improving the Effectiveness of Groups. *Decision Sciences*, 5, p. 459-480.

[14] Hall, J. & W. Watson (1970). The Effects of a Normative Intervention on Group Decision-Making Performance. *Human Relations*, 23, 4, p. 299-317.

[15] Haythornthwaite, C. (1999). Collaborative Work Networks Among Distributed Learners. Hawaii International Conference on System Sciences, Maui, Hawaii.

[16] Hickey, A. M., D. L. Dean, et al. (1999). Setting a Scenario for Collaborative Scenario Elicitation. Hawaii International Conference on System Sciences, Maui, Hawaii.

[17] Hirokawa, R. Y. & D. S. Gouran (1989). Facilitation of Group Communication: A Critique of Prior Research and an Agenda for Future Research. *Management Communication Quarterly*, 3, 1, p. 71-92.

[18] Hoffer, J. A., J. F. George, et al. (1998). *Modern System Analysis & Design*. New York, Addison-Wesley.

[19] Hoffman, L. & N. Maier (1959). The Use of Group Decision to Resolve a Problem of Fairness. *Personnel Psychology*, 12, p. 545-559.

[20] Jarvenpaa, S. L., K. Knoll, et al. (1998). Is Anybody Out There? Antecedents of Trust in Global Virtual Teams. *Journal of Management Information Systems*, 14, 4, p. 29-64.

[21] Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. London, Collier MacMillan.

[22] Kayser, T. A. (1990). *Mining Group Gold: How to Cash in on the Collaborative Brain Power of a Group*. El Segundo, CA, Serif Publishing.

[23] Kelly, G. G. & R. P. Bostrom (1998). A Facilitator's General Model for Managing Socioemotional Issues in Group Support Systems Meeting Environments. *Journal of MIS*, 14, 3, p. 23-44.

[24] Klein, H. K. & M. D. Myers (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* forthcoming(special issue).

[25] Kumar, K. & L. P. Willcocks (1996). Offshore Outsourcing: A Coutry Too Far? European Conference on Information Systems, Lisbon, Portugal.

[26] Maier, N. & R. Maier (1957). An Experimental Test of the Effects of Developmental vs. Free Discussions on the Quality of Group Decisions. *Journal of Applied Psychology*, 41, 5, p. 320-323.

[27] Melymuka, K. (1997). Virtual Realities. *Computerworld*. 31: 70-72.

[28] Musgrave, J. & M. Anniss (1996). *Relationship Dynamics: Theory and Analysis*. New York, NY, The Free Press.

[29] Ngwenyama, O. K. & A. S. Lee (1997). Communication Richness in Electronic Mail: Critical social Theory and the Contextuality of Meaning. *MIS Quarterly*, 21, 2.

[30] Pawar, K. S. & S. Sharifi (1997). Physical or Virtual Team Collocation: Does it Matter? *International Journal of Production Economics*, 52, 3, p. 283-290.

[31] Putman, L. (1986). *Conflict in Group Decision-Making. Communication and Group Decision-Making*. R. Y. Hirokawa & M. S. Poole. Beverly Hills, CA, Sage Publications.

[32] Strauss, A. & J. Corbin (1990). *Basics of Qualitative Research*. Newbury Park, CA, Sage Publications.

[33] Tjosvold, D. & M. M. Tjovold (1995). *Psychology for Leaders: Using Motivation, Conflict, and Power to Manage More Effectively*. New York, NY, John Wiley & Sons.

[34] Tuckman, B. (1965). Developmental Sequence in Small Groups. *Psychological Bulletin*, 3, p. 384-399.

[35] Ven, A. V. d. & A. Delbecq (1974). The Effectiveness of nominal, Delphi, and Interacting Group Decision Making Processes. *Academy of Management Journal*, 17, p. 605-621.

[36] Volkema, R. (1983). Problem Formulation in Planning and Design. *Management Science*, 28, 6, p. 639-652.

A Discussion on Process Losses in GSS: Exploring the Consensus Gap

W. Benjamin Martz, Jr.
Department of Accounting and Information Systems, College of Business, California State University, Chico, USA
Phone: (530) 898-4623, Fax: (530) 898-4970
E-mail: bmartz@cschico.edu

Keywords: Group Support Systems, Consensus Building, Group Problem Solving, Process Losses in Groups

Edited by: Gary Klein

Received: June 1, 1999 **Revised:** October 10, 1999 **Accepted:** December 6, 1999

As research with group support systems (GSS) moves forward, researchers must watch for and identify possible derivative process losses: proposed here as those process losses introduced into the group meeting process while researching a primary dysfunction. This paper reviews a set of GSS literature in order to find support for such derivative process losses. One such loss, "stronger identification with non-consensus," is discussed in more detail. A research study is proposed to help expose this consensus gap in electronic meetings.

1 Introduction

> The fundamental triangle of societal problem solving requires that a team, working with appropriate methodology, address the issue to be resolved. Because of the extensive differences between an individual and a group, it would be astonishing if the same methodology of exploration or inquiry would be equally effective for individuals and groups. (Warfield 1989)

As research in group support systems (GSS) moves forward, it must recognize that the environment in which the research is conducted changes (Briggs et. al. 1998, Nunamaker et. al. 1997). Essentially this means that as researchers identify and study one area of interest, other areas of interest may become salient. The complexity of the group research area almost compels such trade-offs. McGrath characterizes these trade-offs as "horns of a research dilemma" (McGrath et. al. 1982).

The fundamental reason for problem solving teams or groups is to "address the issue to be resolved." As the team works toward resolving that issue though, characteristics of the group members combine with those of the task in what is almost an infinite number of ways of what Simon (1976) refers to as "choice making." Combinations that move groups toward "better" choices or decisions are termed process gains. Those combinations that move the group away from a "better" choice or decision are termed process losses. Shaw (1981) identifies the major areas of process losses and process gains along with significant group research in those areas.

Process losses are found with traditional groups, so researchers, should openly expect to find new process losses identified with electronic groups. As ongoing iterations of research in this area occur that compare manual to electronic environments (Dennis et. al. 1997, Benbasat and Lim 1993), new environments are created. One such environment is the group support systems environment which has been defined as an "interactive, computer-based environment that support[s] concerted and coordinated team effort toward completion of joint tasks" (Nunamaker et. al. 1997).

Adapting Warfield's opening quote to this situation then, why would we expect the manual rules to follow through into the electronic environment. In fact, we should expect these new environments create their own potential new process losses for study. For our purposes, we shall refer to these type of process losses that occur as a consequence of a GSS as "derivative losses."

The work comparing nominal group brainstorming to group brainstorming demonstrates one such evolution of a derivative loss. The format proposed by Osborn (1963), group brainstorming (members producing a list as a group), has been compared over three decades in the research to the nominal group technique (NGT - individuals first working separately then combining to produce a list (Van deVen and Delbecq 1974)) with the NGT winning most comparisons (McGrath 1984). GSS researchers (Dennis et. al. 1993, Valacich et. al. 1994, Nunamaker et. al. 1997), have updated this research stream into the electronic medium where electronic group brainstorming, based on Osborn's brainstorming and Van Gundy's (1981) brainwriting, outperforms its electronic nominal group counterpart. Interestingly, it should be noted that at the same time the brainstorming debate may be moving toward resolution, a new dysfunction - the stronger identification of non-consensus (Benbasat and Lim 1993) may have been introduced.

Research in other areas may lead to similar evolution in those areas. Gallupe et. al. (1992) studied the effects of group size in the electronic and non-electronic environments. Analysis of the results showed that electronic groups evaluated better on productivity and produc-

tion blocking. Shepherd et. al. (1996) studied the impact of social comparison on group performance. Other dramatically broad research questions still abound in this field including questions on anonymity, "flaming," and business process reengineering (Briggs et. al 1998).

So, it is proposed that as GSSs are implemented, researched and used, the new environment may create their own set of group process losses. This paper lists a few candidates for GSS derivative process losses in Table 1, then proposes ground rules for minimizing them in practice. For example, one such loss - stronger identification of non-consensus - is suggested along with some experiential information on minimizing the loss.

2 Potential Process Losses

2.1 Channel Conflict

Most researchers agree that conceptually the activities performed by groups can be defined as either task-oriented or social-oriented. If we envision the two types of activities as information channels and we define the way to get information distributed to a group as the medium, then for traditional meetings with only one serial, verbal medium, we can easily see that there is a conflict for use of the medium by the two channels.

In fact, several methods and scales have been developed to encode a group's activities and to study groups along these scales. Poole (1983) produces meeting flowcharts describing the levels of each activity. Lim and Benbasat (1993) defined meeting comments as "on-task" or interpersonal for their research purposes. In this context, then we have a mis-appropriation of the channel and have created a perceived "loss" when a comment is not "on task."

Miranda and Bostrom (1994) recognize these separate channels in their work on managing group member conflict. In their work, issue-based information is task channel oriented while interpersonal information relates to the social channel. The conflict between the channels was monitored and used in their definition of productive groups. In addition, popular theories when applied to GSSs, position themselves to account for varying levels of both types of activities (see below).

2.2 Information Overload

Review of early problem solving literature (Polya 1957, Warfield 1989, Osborn 1963, Whiting 1958) identifies four generalized problem solving processes or activities: discovery, the uncovering of information; analysis, the decomposing of information into data and perspective; synthesis, the recombining of data into information; and choosing, the act of selection a solution to the problem. These four processes and their relation to the problem domain can be represented using Figure 1.

The divergent processes (uncovering, analysis) historically have been "easier" for groups to accomplish. Research shows that electronic GSS have been able to outperform traditional methods for producing numbers of comments and numbers of unique comments (Shepherd et. al. 1995, Gallupe et. al. 1992, Dennis et. al. 1993, Jarvenpaa at. al. 1998, Lim and Benbasat 1993). Along with this increased production, comes the associated dysfunction of groups inefficiently combining and filtering the large lists of comments, ideas, items, etc.

In fact, this activity of synthesizing large set of information leads directly to one operational definition of task complexity by Benbasat and Lim (1993). Here either of the independent activities of "generating OR choosing" were defined as low complexity, while the combinatory activity of "generating AND choosing" was defined as a high complexity task. Interestingly, this productivity increase has been studied in relation to the amount of conflict it adds to the group (Miranda and Bostrom 1994).

In traditional groups, facilitator and meeting techniques have been used to successfully combat this problem of information overload (Doyle and Strauss 1982, Bradford 1976). The practicality of the facilitator in the role of information manager has been studied in electronic meetings (Griffith et. al. 1998). There are indications that GSSs are allowing groups to produce more information than they or the GSS can economically, effectively or efficiently handle. This calls for better and more techniques in GSSs to address the synthesis activity.

2.3 Overhead Costs

Meetings have overhead costs. Dennis and Valacich (1993) propose that one such cost is the time "participants in electronic groups must read and think about ideas before those ideas can stimulate new ideas." Diehl and Stroebe (1991) are a little more harsh when they proclaim that from their research with idea producing groups, "waiting time is not used productively." Without passing judgment on the specific positions, it will be stipulated that there exists inefficiencies in having the whole group wait while one member completes the reading of the accumulated information.

2.4 GSS Implies Wrong Structure

There are numerous methodologies and techniques for working with groups. For example, one can find over 100 such techniques used by Fortune 500 businesses in the Vest Pocket CEO (Hiam 1990). Automating these techniques leads to an interesting dilemma though. Each of the automated techniques comes with an inherent bias which may or may not be appropriate for the group's problem environment. The ways in which these methodologies and techniques organize information and permit groups to exchange that information is referred to as the "communication structure" (Johnson 1993). Two popular theories try to incorporate this notion of communication structure into research models for GSSs. Both are concerned with how well the group environment and GSS technique match.

Table 1: GSS Process Gains and Losses

Derivative Process Losses	Primary Process Gains
channel conflict	Better analytical support
information overload	Easier multi-phase voting
overhead costs	More reflective
GSS influence choosing wrong "structure"	Increase in "effective" group size
stronger identification of non-consensus	Wider perspective of information domain
	Removal of time and geographical constraints

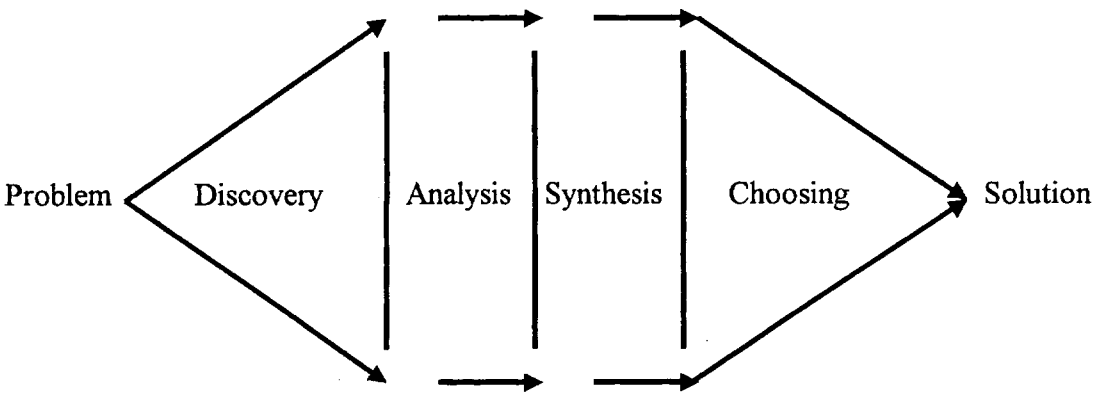


Figure 1: General Problem Solving Activities/Phases

Media Richness (Dennis et. al.) places different group environments such as face-to-face, telephone and written environments along a spectrum of how much of these activities (social and task) the environment can support. It further proposes that we need to make sure the environment is large enough to accommodate the needs of the group, the task and the GSS technique. Those groups which do not have a media rich enough (large enough) channel are susceptible to varying degrees of failure. Therefore, the GSS chosen precludes the appropriate technique from being selected by imposing too narrow a structure.

Adaptive Structuration Theory (AST) outlined by Poole and DeSanctis (1989) undertakes to explain how groups adapt to new environments, especially as technology is introduced. A concept of "appropriation" is used in the theory to refer to "the manner in which structures are adapted by a group for its own use through a process called structuration, wherein structures are continuously produced and reproduced (or confirmed) as the group's interaction process occurs" (Gopal et. al. 1993). Here a group, through successful structuration, can use the channel for both social and task oriented activities.

In all cases, the wrong techniques or meeting structures - those not providing enough richness or those not matching the group's environment - can be chosen simply because they are the ones available in a GSS. This is the electronic version of Groupthink, the dysfunctional group process of myopic thinking, where groups continue down an inappropriate meeting activity without questioning and are encouraged by the structure bias of a GSS.

2.5 Reduction in Consensus

The American Heritage Dictionary defines consensus as 1.) collective opinion 2.) general agreement or accord. Looking closer at this concept, one observes that consensus is an opinion and therefore a human characteristic. In turn, this opinion creates a position or perspective. Finally, the concept of "general agreement" can be viewed as a harmony within some defined tolerances. Once the tolerance of agreement is broken, then there is no consensus.

Juxtaposed against what consensus means, there are at least three important concepts of what consensus does NOT mean. First, it does not mean "no disagreement." In fact, the above definition explicitly allows for disagreement with the concept of tolerances. Second, in some situations consensus may or may not be a goal of the group. Finally, the level of consensus is not static and may change quickly based upon new information received by the group.

Individuals hold the dictionary and many more fundamental definitions of consensus when they combine into groups. Any definition of consensus for the group must respect an amalgam of these individual perspectives. But, in the end, it is this amalgam which generates the derivative process loss or dysfunction in the group. For example, one individual may believe that a meeting produced a high level of consensus while a second individual believes little consensus was achieved in the same meeting.

A comprehensive definition of consensus remains elusive with some researchers prescribing consensus while others are willing to simply describe. On the one side, Sniezek and Henry (1990) calculate consensus using "judgment accuracy" in their studies on consensus and corresponding social interaction. Bradford (1976) simply describes consensus seeking as "the maintenance function" of a meeting that resolves the polarization which occurs around issues in meetings. From here, he lets the facilitator and group define consensus for the specific task.

Compiling these concepts, the following working definition for group consensus is proposed:

> Group Consensus [on a topic] is a group position to which the members of a group have contributed and about which the members are in general agreement or accord.

Benbasat and Lim's meta-analysis (1993) of experimental studies with group support systems (GSSs), evaluated group support research studies based upon the studies' own definitions of consensus. Interestingly, Benbasat and Lim identify a "reduction in consensus" and call for additional exploration in this area.

There are three major observations relating to group consensus which may help explain how a "reduction in consensus" may be generated by working in a GSS. First, creating group consensus seems to be a fundamental activity to group problem solving. Second, developing a group consensus is a "high complexity" task, implying difficulty for groups to successful complete. Third, developing consensus implies an inherent process conflict.

Most processes or methodologies which wish to "resolve an issue" do so in the effort to gain a choice or consensus about a choice. Therefore most problem solving methodologies have the fundamental activity of consolidating individual perspectives into a group perspective in order to choose or create a "best answer." For example, Churchman's alternative assessment (1979), Mason and Mitroff's stakeholder assessment (1981), Saaty's priority scaling models (1980), and Fox's voting methods (1987) all address verifying alternatives presented by group members before choosing.

As we see in Table 2, many authors have divided problem solving into two sub-processes geared to divergent and convergent activities. These and their counterparts are used by Benbasat and Lim in their meta-analysis to define task complexity. In their work, a task which undertakes both processes would be high complexity. Often developing consensus requires these two sub-processes and under this definition easily should be accepted as a highly complex group task.

These first two observations lead to the third; and to what would seem to be a paradox of consensus. The sub-processes found in consensus-building antagonize each other. There exists a significant tension between divergent and convergent processes. The paradox is that identifying this tension or conflict is necessary to build the "agreement or accord." Too much tension is chaos and anarchy; too little tension is groupthink.

The tension is exposed by reviewing Osborn's (1963) and Whiting's (1958) brainstorming rules while knowing that in the near future, the group must categorize any list created using those rules. The rules: "Judgment is ruled out," "free-wheeling is encouraged," "Quantity is desired," "Use a broad experience base of participants," "Include an 'outside.' member," when applied to group brainstorming produce a list of items that resists categorization. It is this resistance that we may be identifying and labeling as non-consensus when, in reality, we have as much functional consensus - the level of consensus group members perceive - as we have ever had.

In summary, successful consensus building is a highly complex task because it must interface two or more diametrically opposed meeting sub-processes. If GSSs help groups produce more in the divergent process and but do not help in the convergent process, then less consensus should be expected.

3 Research Model

An exploratory research study was undertaken to explore the derivative loss of non-consensus observed by GSS researchers. It was proposed that electronic groups realistically get more information than verbal and it is this additional level of information that leads to non-consensus. We believed that there were two sources of this information unique to the electronic environment when compared to the verbal environment: the discussion process and the results of the vote process. An experiment was designed and implemented in order to test these two areas.

Twenty-two, 10-person group meetings were run over a two-week period at the end of a semester. While specific demographics were not kept, the subjects represent the general student-body characteristics within the College of Business. The College itself provides a residential campus environment in a small, state-supported university climate. Participants were full-time, junior and senior level students from five different courses. The student's ages ranged from early twenties, as with traditional students, to late forties, found with some re-entry students. The task for the meeting was a discussion and rank-order vote regarding five methods that faculty use to evaluate students. Two rewards for the participants were associated with the meeting. One reward that went to all participants was extra credit points

Table 2: Problem Solving Sub-processes

Researcher	Divergent Process	Convergent Process
Osborn(1963)	Ideation	Synthesis
Cowan(1986)	Clarification	Categorization
Polya(1957)	Decomposing	Recomposing
DeBono(1985)	Lumpers	Splitters
Warfield(1989)	Analysis	Synthesis
Benbasat & Lim(1993)	Generating	Choosing

in their class. The second was a chance to win $10 cash by "focusing" on the task.

Each group experienced one of four meeting formats: Electronic Unknown (EU); Electronic Known (EN); Verbal Unknown (VU); Verbal Known (VN). A handout was developed that described the concept of consensus and how to measure the level of agreement between rank-ordered lists from multiple judges. This handout was pre-tested and reviewed for clarity and understanding with students, who were not candidates for the study. The same handout and narrative was presented to all group members in all groups. During the presentation, the acceptable level of consensus was defined as the "level of consensus, between 0.0 and 1.0, at which you, personally, would say a group has consensus." Each participant was asked to record his/her level of acceptance (ACC) for an "average" student group.

All groups used the same electronic meeting room for the purposes of the maintaining the same meeting environment. The five methods (Tests, Homework, Quizzes, Term Paper, Group Project) were presented to each group. A definition of each method was read to the participants and any questions for clarification answered. The participants were instructed to discuss the five methods amongst themselves. Half the groups used electronic software (E) and half used traditional verbal discussion (V). Electronic discussions were implemented using the GroupSystems software and no verbal discussion allowed. Groups moved on to voting when their discussions ended. Discussions ended once the group members agreed verbally to move on; either when typing and reading stopped in the electronic groups or when the conversation stopped in the verbal groups. Discussions varied in length from 8-12 minutes.

After the discussions, each participant was asked to rank order the list of grading methods based upon the instruction "Which method, do you feel, represents the best way for a student to accurately display that he or she knows a topic. The 'best' method would go to the top of the list; followed in order by the next best until all items on the list have been ordered." All groups used electronic software to record their votes. After the votes were collected, but before the results were displayed, each participant was requested to estimate the level of consensus (EST). Referring back to the level of consensus discussion, each participant recorded a number between 0.0 and 1.0 for the discussion and vote they just performed.

The results were displayed to the group in matrix format similar to that used in the handout. Half of the groups received the full matrix showing the actual (ACT) computed consensus and therefore knew (N) the actual consensus for their group from that point forward. The other half of the groups received the full matrix but without the actual calculated consensus displayed and therefore the actual computed consensus remained unknown (U) to them from that point forward.

Each participant completed a questionnaire. Five of the questions (Q1 through Q5) were geared toward their perception of the outcomes of the meeting. One question each to cover; satisfaction with the outcome of the meeting (Q1), satisfaction with the process of the meeting(Q2), satisfaction with fair representation in the discussion portion of the meeting(Q3), satisfaction with the fair voting portion of the meeting(Q4), satisfaction with the use of time in the meeting(Q5). Four additional questions were used to ask the students about: how seriously they took the exercise(Q6), the importance of the rewards(Q7), ease of use for the voting software (Q8) and when used, the ease of use for the discussion software(Q9).

The actual level of consensus was revealed to the rest of the groups. The $10 reward was awarded to that participant who had estimated the closest to the actual group consensus. The participants were debriefed as to the purpose of the study and admonished not to share the purpose or content of the experiment with other students who may yet be part of the experiment.

4 Results

The difference in acceptable (ACC) levels of consensus across the groups proved insignificant thereby establishing that the research was starting with similar groups with respect to their pre-experiment characteristic – ACC levels.

In addition, many of the results were consistent with literature on which the conjectures were based. Groups with verbal discussions showed a higher level of actual consensus (ACT) than did the groups with electronic discussion. From the questionnaire results, the saliency of the rewards and ease of use seemed strong and thus would not act as confounds. With regard to the discussion process, participation was higher in electronic groups when measured by number of comments and by number of participants who contributed to the discussion. All of these findings indicate

that we had created an environment consistent with those that had produced non-consensus in the past and on which we can appropriately perform our tests.

Overall, the differences between question 1(Q1), question 3(Q3) and question 4(Q4) were all significant. The difference in group consensus among the four groups was significant (.0191). Differences in actual group consensus (ACT) between the groups in the electronic discussion (EN and EU) and the groups in the verbal discussion (VN and VU) carried a lower level of significance (.0893 and .0637 respectively). Significant differences were also found among the estimates (EST) made by the participants in the four groups.

Several differences between the verbal and electronic discussion groups proved significant. Verbal groups were more satisfied, defined as more willing to accept, their meeting outcomes (Q1). Paradoxically, Electronic groups were more satisfied than their verbal counterparts with two of the meeting components: their representation in the meeting (Q3) and their voting process (Q4). No significant differences were found between the known – unknown groups for any of the five questions.

5 Discussion

The study provides additional evidence of what may be termed a consensus gap for groups. There is a significant difference in the estimates of consensus (EST) between Verbal and Electronic groups. In addition to having less actual consensus, participants in electronic groups also estimate their consensus lower than those in verbal groups. So, their perceived non-consensus (ACC – EST) is greater. We know that they feel that their meeting process is "more fair" in the area of discussion and voting, but they are less satisfied with the meeting outcomes than the verbal groups. Based upon our findings, the fact that the group knows or does not their actual consensus does not seem to impact their satisfaction with the meeting outcome or process.

We must look deeper into the differences between the electronic and verbal discussion processes to find an explanation for the higher levels of non-consensus in electronic discussions. One hint we have is that the participants observe and discount their estimates of consensus more if they are in an electronic environment. Again, as more information is shared and more participation occurs, this derivative loss become more plausible.

The importance of this derivative loss phenomenon to the practical world is growing. We are anticipating the practical use of virtual work groups in the business world, in the academic world and for leisure time. As pointed out here, one major concern will be how well virtual groups will be able to institute, monitor and determine consensus in their decision making.

6 Summary

In summary, this paper proposes that as GSSs are created and targeted toward addressing process losses in traditional meeting and group environments, new types of process losses may be generated. These are derivative - or second generation - losses brought on by the interaction between the historical variables of group and task characteristics and the new environment of GSS. Five such derivative process losses were identified from the literature and discussed. The paper concluded with a summary of a research project that studied one of those derivative losses – more non-consensus in electronic groups. The results of the study reiterated the findings in the literature concerning comparisons between electronic and verbal groups. In addition, it proposes that the real issues for increased non-consensus reside in the differences in the discussion or production of information in meeting.

References

[1] Bradford, Leland P. (1976) *Making Meetings Work*, University Associates.

[2] Briggs, Robert O., Jay F. Nunamaker, Jr. & Ralph H. Sprague, Jr. (1998) 1001 Unanswered Research Questions in GSS, *Journal of Management Information Systems*, 14, 3, p. 3-21.

[3] Brightman, Harvey J. (1988) *Group Problem Solving*, Georgia State University Press.

[4] Benbasat, Izak & Lai-Huat Lim (1993) The Effects of Group, Task, Context and Technology Variables on the Usefulness of Group Support Systems, *Small Group Research*, 24, 4.

[5] Churchman, C. West (1979) *The Systems Approach*, Laurel.

[6] Cowan, David A. (1986) Developing a Process Model of Problem Recognition, *Academy of Management Review*, 11, 4, p. 763-776.

[7] DeBono, Edward (1985) *DeBono's Thinking Course*, Facts on File Publications.

[8] Dennis, Alan R. & Susan T. Kinney (1998) Testing Media Richness Theory in the New Media: The Effects of Cues, feedback and Task Equivocality. *Information Systems Research*, 9, 3, p. 256-274 .

[9] Dennis, Alan R., Jay F. Nunamaker, Jr. & Douglas R. Vogel (1991) A Comparison of Laboratory and Field Research in the Study of Electronic Meeting Systems. *Journal of Management Information Systems*, 7, 3, p. 107-135.

[10] Dennis, Alan R. & Joseph S. Valacich (1993) Computer Brainstorms: More Heads are Better Than One, *Journal of Applied Psychology*, 78, 4, p. 531-537.

[11] Diehl, Michael & Wolfgang Stroebe (1991) Productivity Loss in Idea-Generating Groups: Tracking Down the Blocking Effect, *Journal of Personality and Social Psychology*, 61, 3, p. 392-403.

[12] Doyle, Michael & David Strauss (1982) *How to Make Meetings Work*, Jove edition.

[13] Fox, William M. (1987) *Effective Group Problem Solving*, Jossey-Bass.

[14] Gallupe, R. Brent, Alan R. Dennis, William H Cooper, Joseph S. Valacich, Lana M. Bastianutti, & Jay F. Nunamaker, Jr. (1992) Electronic Brainstorming and Group Size, *Academy of Management Journal*, 33, 2, p. 350-369.

[15] Gopal, Abhijit, Robert P. Bostrom, & Wynne W. Chin (1993) Aplying Adaptive Structuration Theory and Group Support Systems Use, *Journal of Management Information Systems*, 9, 3, p. 45-69.

[16] Griffith, Terry L., Mark A. Fuller & Gregory B. Northcraft (1998) Facilitator Influence in Group Support Systems: Intended and Unintended Effects, *Information Systems Research*, 9, 1, p. 20-36.

[17] Hiam, Alexander (1990) *The Vest-Pocket CEO*, Prentice-Hall.

[18] Jarvenpaa, Sirkka L., V. Srinivasan Rao & George P. Huber (1988) Computer Support for Meetings of Groups Working on Unstructured Problems: A Field Study, *MIS Quarterly*, p. 645-666.

[19] Johnson, J. David (1993) *Organizational Communication Structure*, Ablex Publishing.

[20] Jones, Jack William, Carol Saunders & Raymond McLeod, Jr. (1989), Information Media and Sources Across Management Levels, *Journal of Management Information Systems*, 5, 3, p. 71-84.

[21] Kelly, Gigi G. & Robert P. Bostrom (1998) A Facilitator's General Model for Managing Socioemotional Issues in Group Support Systems Meeting Environments, *Journal of Management Information Systems*, 14, 3, p. 23-44.

[22] Lim, Lai-Huat & Izak Benbasat 1993) A Theoretical Perspective of Negotiation Support Systems, *Journal of Management Information System*, 9, 3, p. 27-44.

[23] Maier, Norman R. F. & Allen Solem (1952) The Contribution of a Discussion Leader to the Quality of Group Thinking: The Effective Use of Minority Opinions, *Human Relations.*

[24] Mason, Richard O. & Ian I Mitroff (1981) *Challenging Strategic Planning Assumptions*, John Wiley & Sons.

[25] Massey, Anne P. & Danial L. Clapper (1995) Element Finding: The Impact of a Group Support System on a Crucial Phase of Sense Making, *Journal of Management Information Systems*, 11, 4, p. 149 - 176.

[26] McGrath, Joseph E., Joanne Martin & Richard A. Kulka (1982) *Judgment Calls in Research*, Sage Publications.

[27] McGrath, J. E. (1984) *Groups: Interaction & Performance*, Englewood Cliffs, NJ, Prentice-Hall.

[28] Miranda, Shaila M. & Robert P. Bostrom (1994) The Impact of Group Support Systems on Group Conflict and Conflict Management, *Journal of Management Information Systems*, 10, 3, p. 63-96.

[29] Nunamaker, Jr., Jay F., Robert O. Briggs, Daniel D. Mittleman, Douglas R. Vogel, & Pierre A. Balthazard (1997) Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings, *Journal of Management Information Systems*, 13, 3, p. 163-207.

[30] Osborn, Alex F. (1963) *Applied Imagination*, Scribners.

[31] Poole, M.S. & G. Desanctis (1989) Understanding the Use of Group Decision Support Systems: The Theory of Adaptive Structuration. In *Theoretical Approaches to Information Technologies in Organizations*, C. Steinfeld and J. Fulk, eds. Beverly Hills, Ca., SAGE Publications.

[32] Poole, Marshall Scott (1983) Decision Development in Small Groups II: A Study of Multiple Sequences in Decision Making, *Communications Monographs*, 50.

[33] Polya, George (1957) *How to Solve It*, 2nd ed., Princeton University Press.

[34] Saaty, Thomas L. (1980) *The Analytical Hierarchy Approach*, McGraw-Hill.

[35] Shaw, M. E. (1981) *Group Dynamics: The Psychology of Small Group Behavior*, 3rd. ed., New York, New York, McGraw-Hill.

[36] Shepherd, Morgan M., Robert O. Briggs, Bruce A. Reinig, Jerome Yen & Jay F. Nunamaker, Jr. (1995-96) Social Comparison to Improve Electronic Brainstorming, *Journal of Management Information Systems*, 12, 3, p. 155-170.

[37] Silver, Mark S. (1988) User Perceptions of DSS Restrictiveness, *Journal of Management Information Systems*, 5, 1, p. 51-66.

[38] Simon, H.A. (1976) *Administrative Behavior: A Study of Decision-Making Process in Administrative Organization*, 3rd ed., New York, Free Press, p. 40.

[39] Sniezek, Janet A. & Rebecca A. Henry (1990) Revision, Weighting and Commitment in Consensus Group Judgment. *Organizational Behavior and Human Decision Processes*, 45, p. 66-84.

[40] Valacich, Joseph S., Alan R. Dennis & Terry Connolly (1994) Idea Generation in Computer-based Groups: A New Ending to an Old Story. *Organizational Behavior and Human Decision Processes*, 57, p. 448-467.

[41] Van deVen, Andrew H. & Andre L. Delbecq (1974) The Effectiveness of Nominal, Delphi and Interacting Group Decision Making Processes. *Academy of Management Journal* 17, 4.

[42] Van Gundy, A.B. (1981) *Techniques for Structured Problem Solving*, New York, Van Nostrand Reinhold.

[43] Warfield, John N. (1989) *Societal Systems*, Intersystems Publications.

[44] Whiting, Charles S. (1958) *Creative Thinking*, Reinhold Publishing.

Factors Affecting the Use, Adoption and Satisfaction with Groupware

Marion G. Sobol
Southern Methodist University, E.L. Cox School of Business, Dallas, Texas 75275, USA
Phone: 214 768-3171, Fax: 214 768-4099
E-mail: msobol@mail.cox.edu
AND
Mary Anne Winniford
Enterprise Management Associates, 5398 Manhattan Circle, Boulder, Colorado 80303, USA
E-mail: winniford@enterprisemanagement.com

Keywords: Groupware, Group Support Systems, Satisfaction, Functionality, Adoption

Edited by: Gary Klein
Received: June 1, 1999 **Revised:** November 11, 1999 **Accepted:** December 6, 1999

This survey reports on corporate use of groupware, specifically Lotus' Notes, Novell's Groupwise and Microsoft's Exchange. Out of 71 businesses that responded, 82% use groupware. The functions ranked most important and most common still focus on traditional communication such as knowledge sharing, remote access and e-mail. Newer functions, such as customer service, launching a website or E-commerce platform, reported much lower importance ranking and usage. Overall, groupware received high ratings, including overall satisfaction, product satisfaction and increasing productivity. The respondents included both IT managers and end-users, who responded similarly, except that managers were somewhat more critical of the groupware. The MIS department led the adoption process in terms of where the respondents learned about groupware, proposals for adoption of groupware, and sponsorship of groupware adoption. Some differences were found in software adoption by sponsor: top management preferred Lotus' Notes, and sponsored larger installations.

1 Introduction

Developments began in the mid-1980s that changed the way businesses viewed computing. Instead of one person working at one computer, new computer support for groups working together were evolving. Many factors contributed to this including "... a) computation inexpensive enough to be available to all members of some groups; b) a technological infrastructure supporting communication and coordination, notably networks and associated software; c) a widening familiarity with computers, yielding groups willing to try the software; d) maturing single user application domains that pushed developers to seek new ways to enhance and differentiate products." (Grudin, 1994). These tools were called group decision support systems (GDSS) or computer-supported cooperative work, (CSCW) depending on the type of groups they supported. GDSS tools were used to support meetings where all the participants met at the same place and the same time. CSCW tools could also support groups meeting at the same time, although in dispersed locations. Electronic mail was becoming common in academic environments, supporting communication between people across time and space. This communication could be either synchronous or asynchronous. The more all-encompassing term "groupware" came about in the late 1980s. A complete definition of groupware is provided by S. R. Ahuja (1989): "Groupware refers to computer and communications systems that provide communication, coordination and collaboration services for cooperating people." J. R. Ensor (1990) provides a common, simpler definition: "Computer-based systems that help two or more people work together are called groupware."

Just what does "communication, coordination and collaboration" mean in terms of functionality? Several companies are trying to determine what the customer wants and working to develop it as fast as possible. Three big competitors are Lotus' Notes, Microsoft's Exchange (Outlook), and GroupWise by Novell. These products have split the market pretty evenly. Smaller start-up companies have also entered the groupware market (Gagne, 1997). This area is growing and still being defined. New functionality is constantly being added. Some of the software changes include use of Java, and Internet capabilities (Radosevich, 1997; Lavilla, 1997), incorporation of workflow (Bort, 1997), and use of standardized scheduling tools (Thomas, 1997). To date, there is little research to guide this development, in terms of which of these capabilities are being used, how important they are, and how satisfied people are with their groupware. Many academic research studies have been done on EMS (electronic meeting systems) but most of these studies have ignored asynchronous software such as Lotus Notes, Microsoft's Exchange and Novell's Groupwise (Pervan, 1999). By far the latter packages are the most commonly used softwares in business settings.

Many corporations have implemented some kind of groupware technology. Many of the large consulting firms such as Andersen Consulting or Ernst & Young have invested a lot of money in groupware (Garcia,1997). At first the focus was within the IT arena, but businesses have determined that this technology can be used in many areas of business. Our recent survey of 71 companies listed by Computerworld as most efficient users of computers indicated that 82 percent were using groupware. There has been little indication of how that technology has been adopted, or how it has spread from the IT department to the rest of the corporation.

In 1999, Downing and Clark examined the expectations and realization of benefits of the use of groupware for 22 consulting firms. Although, they had high expectations concerning groupware implementation in terms of increased customer service, increased communication, competitive advantage, cost savings, higher productivity and leveraging of expertise of the consultancy, they found varying degrees of success. They generally attributed good results to a sharing culture, highly visible champions and ability to set realistic benefit expectations.

2 Purpose of This Study

Most of the recent studies of groupware have focused on in-depth case studies of how and why the groupware was adopted and how it was used. Orlikowski (1997) showed an improvisational model for change management using Lotus Notes to develop an incident tracking support system. Another important stream of in-depth research has dealt with ways of ensuring that people actually conform to the appropriate decision support heuristics suggested by their groupware (Nunamaker et al. 1991; Hirokawa and Rast 1992; DeSanctis and Gallupe, 1987; Wheeler and Valacich 1996).

There are very few statistical studies of satisfaction with wide-spread commercial groupware, the importance of various functions, and sponsorship of groupware adoption. This is probably due to the fact that in previous years widely different groupware has been developed for specific applications like bank credit decisions and make-buy-rent decisions. It is only recently that more universal groupware packages such as Lotus Notes, Microsoft Exchange and Novell's GroupWise have been developed. These packages focus on the communication aspects but can be developed to include decision support heuristics.

Since these groupware packages have been more universally adopted it now becomes possible to perform cross-sectional surveys on the adoption, use and satisfaction with these packages. In 1997, Gagne interviewed 25 groupware users at Fortune 500 companies and published the finding in *Computerworld*. This study was able to state only non-numerical highlights because of small sample size. Moreover, it didn't include any information on how people had heard of software. Since mass groupware is relatively new, few other numerical studies have been made to evaluate the use of these tools.

3 Methodology

In our study we have polled IS managers from the Computerworld lists of the best users of IS technology 1994-5. In addition we have polled IS users and the department managers from large companies throughout the U.S. The final sample was 71 with 58 or 82 percent indicating that they used one of the three major forms of groupware - Lotus Notes, Microsoft Exchange or Novell's Groupwise. The response we received from our 100 top users were primarily from managers (93%) whereas responses from the sample of IS users and department managers from other large companies showed that 33% were managers and 67% were non-managers. This inclusion of non-managers enabled us to make comparisons of managerial and non-managerial views. Of the 57 respondents who were using the groupware, 62 percent were in the MIS department and the remaining respondents were scattered in other departments. The largest non-MIS representation came from finance departments (16%). The sizes of the departments represented varied. A third of the users (33.3%) were in departments of 10 or less, thirty-five percent were in departments of 10-50 people. About twenty-eight percent were in departments that were larger than 50. The remaining 3-4 percent did not state how large their departments were.

The survey was developed based on questions asked in a previous smaller study (Gagne, 1997), however, many additional questions were added as suggested by our pilot study and our own research interests. The questionnaire was pilot tested with managers of a high-tech company. Order and question clarification was modified as a result of their input, and several functions were added to our list of applications.

4 Overall Results

In this study the respondents were asked to name the groupware package they used. As in a previous study, three groupware packages were found to be the most commonly used: Lotus' Notes, Novell's GroupWise and Mircosoft's Exchange (Outlook) (Gagne, 1997). Forty-two percent used Notes; forty-one percent used Microsoft's Exchange, and fourteen percent used Group Wise. About 3% said they were using other software (Table 1a). Forty-one percent had used the software for less than 12 months. Twenty-three percent had used these groupware packages for 12-24 months and the remaining 36% had used this software for more than 24 months (Table 1b).

The number of users on the system tended to be more than 100, with 37.5% having 100 but less than 1200 users and 21.8% reporting more than 1200 users (Table 1c). Only 18.7% reported less than 20 users. For the people who are using groupware, 75% reported that they had training. Most people reported 4 hours or less (65.4%). In fact,

Table 1: Overall Responses

a. Groupware used (58*)	
Lotus Notes	42.4%
Group Wise	13.6
Microsoft Exchange	40.7
Other	3.3
	100.0%
b. How long owned? (58*)	
0 to 12 months	41.5%
13-24 months	22.6
More than 24	35.9
	100.0%
c. Number of users? (32*)	
Less than 20	18.7%
20 – 100 users	21.8
100 – 1200 users	37.5
More than 1200	21.8
	100.0%
d. Received Training? (56*)	
Yes	75.0%
No	25.0
	100.0%
e. How trained? (42*)	
1 hour or less	14.7%
2 Hours	29.4
3 - 4 Hours	20.6
8 Hours	23.5
16 - 24 Hours	6.8
	100.0%
f. Hours of training? (34*)	
Mean training time:	5.24 Hours
In-house	45.3%
Outside	16.7
Combination	38.0
	100.0%

*number of respondents for this question

Table 2: Satisfaction Measures

a. Does it work well? (58*)	percent
Yes	88.7
No	7.5
Yes & No	3.8
	100.0
b. How satisfied are you? (58*)	
Weighted mean	3.84
1-3	28.6
4	53.6
5	7.8
	100.0
c. Do your job better? (58*)	
Yes	88.8
No	5.6
Don't know	5.6
	100.0

d. Satisfaction related to length of Use

Years used	Mean score
0-12 months	3.75
13-24 months	4.00
More than 24	3.82

*number of respondents for this question

the median training time was 3.2 hours. The mean training time was 5.24 hours because there was a small group with 16-24 hours of training. There were however 23.5% who reported 8 hours and another 6.8% had 16-24 hours of training (Table 1d,e). For 45.3% training was in-house, 16.7% were trained outside and 38.0% reported both in house and outside training (Table 1d).

Generally people using these groupware packages thought that they worked well, with 88.7% saying yes it worked well, 7.5% said it did not work well and 3.8% had a mixed reaction (Table 2a). On a five point Likert scale with 5 as very satisfied and one as not satisfied the group rated their satisfaction with the software as 3.84 (Table 2b). People were likely to feel that they could do their job better using groupware (88.8%). About 5.6% felt that the groupware did not improve their job performance and the remaining 5.6% weren't sure (Table 2c). As the time increased mean satisfaction rose, however, the pattern tended to be curvilinear. Most satisfaction was expressed by those using software between 13-24 months (Table 2d).

When the three important groupware packages were compared on the question of "does it work well," they all generated a high level of approval. For the two most popular packages, Lotus Notes and MS Exchange (Outlook), the yes scores were 86.4% and 90.9% respectively (Table 3a). For Novell all users (100%) said it worked well but this was a small sample.

On the overall satisfaction weighted average score, Lotus Notes was 3.875, MS Exchange (Outlook) was 3.909 and Novell's groupware was 3.625 (Table 3b). These scores were not significantly different. When we asked whether these packages had increased productivity 91.3% of Lotus Notes users said yes, 100% of Novell groupware users said yes. A large number of the MS Exchange (Outlook) users did not respond so we couldn't tabulate these results (Table 3c).

The most recently adopted groupware was MS Exchange (Outlook) where 71.4% have had the package for 12 months or less. In contrast 26.1% of Lotus Notes and

Table 3: Satisfaction Measures by Groupware

	Lotus Notes	Novell's Groupwise	MS Exchange
a. Does it work well?			
Yes	86.4%	100.0%	90.9%
No	9.1	-	4.5
Don't know	4.5	-	4.6
	100.0%	100.0%	100.0%
b. Satisfaction Score?			
Weighted Mean	3.88	3.63	3.91
c. Has it increased productivity?			
Yes	91.3%	100.0%	response
No	8.7	-	too small to
Too early to judge	-	-	tabulate
	100.0%	100.0%	
d. When adopted?			
12 months or less	26.1%	14.3%	71.4%
13 - 24 months	30.4	28.6	14.3
more than 24	43.5	57.1	14.3
	100.0%	100.0%	100.0%

14.3% of Novell users report that they have had their software for 1 year or less (Table 3d). The package held longest is Novell's GroupWise. 57.1% report that they have used the package for more than 2 years. Lotus Notes packages have been used for more than 24 months by 43.5% of the Lotus Notes users.

5 Functionality Results

In previous studies, no evaluation of the importance of various functions had been performed so the respondents were asked to rate the importance of various groupware functions and then to state whether they used the groupware for this particular function. Of the thirteen functions listed in the questionnaire seven were used in an earlier Computerworld (Gagne, 1997) survey and the remaining six were derived from our initial pilot study and review of the literature. Table 4 lists the weighted mean importance scores, (with 1 as unimportant and 5 as very important), and ranks the uses in terms of their mean importance score. This table also lists the percent of respondents who use this function and ranks the functions by percent using them.

Importance of the various types of functionality for the DSS were measured using a Likert scale which features an even number of favorable and unfavorable statements to that the scale is balanced. (See questionnaire, Likert 1970) We used a 5-point scale with neutral as 3. A recent study by Maurer and Pierce, 1998 showed that when a Likert type was tested against traditional measures of self efficacy the Likert scale had similar reliability error variance, provided equal levels of predictive ability, and had similar factor structure and similar discriminability.

The most important uses for groupware are knowledge sharing and remote access. Next in line are discussion and collaboration for projects and workflow, scheduling, discussion forums for wide groups and document management. All of these uses have weighted means of 3.29 or better. The remaining uses (ranking from 8-13) are monitoring workflow (2.582), tracking hardware/software inventory (2.453), converting from mainframe to client server (2.420), tracking customer service and purchase orders (2.407), launching a Website (2.320) and building a platform for E-commerce (2.231). Thus group interaction is of crucial importance in using groupware while E-commerce and launching of Websites are not as important here. The percent who used the groupware for each of these functions is very highly correlated with the importance rankings as the rank order correlation coefficient is r = .9011.

Open-ended questions were also asked to find out what functions were useful. We have examined the closed-ended questions about what features of groupware were important, but these open-ended questions unearthed some other important uses (Table 5). Foremost was E-mail which 39.5% found useful. The next most important areas were scheduling 13.1% which was also important in Table 4. Overcoming time zones was mentioned by 12.5% and better integration with internal applications was mentioned by 9.4%. Another open-ended question was, "how could groupware be improved?" (Table 6) The main comments dealt with integration. Better integration with internal applications (15.2%) and with other desk top applica-

Table 4: Applications for Groupware Importance and Percent Who Use Them

Application	Weighted Mean	Rank Order	% Who Use	Rank Order
Knowledge Sharing	4.36	1	96.4	1
Remote Access*	4.05	2	87.7	2
Discussion & Collaboration in Projects*	3.93	3	85.5	3
Workflow	3.68	4	57.9	7
Scheduling	3.64	5	69.6	4
Discussion Forum (Wide Groups)	3.43	6	63.6	6
Document Management	3.29	7	68.4	5
Monitoring Workflow*	2.58	8	28.1	9
Tracking Hardware & Software Inventory*	2.45	9	23.2	11
Converting from Mainframe to Client Server	2.42	10	21.4	12
Tracking Customer Service Requests*	2.41	11	33.9	8
Launching Website*	2.32	12	25.0	10
Building a Platform for E-Commerce*	2.23	13	19.6	13

Rank Order $\rho = .9011$

*Application suggested by Gagne, 1997

Table 5: What Functions Were Useful?

E-mail	39.5
Scheduling	13.1
Overcoming time zones	12.5
Better integration with internal applications	9.4
Address books	5.3
Database users	2.6
Information center	2.6
Monetary receipts & opening mail	2.6
Creating & organizing folders	2.6
Other, don't know	10.5
	100.0

percentages of 38 respondents

Table 6: How Could Groupware Be Improved?

Better integration with internal applications	15.2
Better integration with other desktop applications	12.1
Use more functions	12.1
Easier to use	9.1
Better administrative management	9.1
More staff to develop more applications	6.1
More licenses/users	6.1
Don't know, other	27.3
	100.0

percentages of 33 respondents

tions were mentioned (12.1%). The respondents also suggested that they should use more functions of the groupware (12.1%). Other not as frequently mentioned improvements were: easier to use, better administrative management, more staff to develop more applications and more users (Table 6).

Finally respondents were asked how they intended to extend their groupware next year (Table 7). A little over a third (23) said they would extend it. The main extension was to add more users (30.4%) Others mentioned new applications (17.4%) and conversion to MS Exchange (Outlook) (17.4%). Another 17.4% wanted to encourage more Intranet use or Web integration.

Table 7: How will you extend groupware next year?

Extension	percent
More users	30.4
New applications	17.4
Convert to MS Exchange	17.4
Encourage more Intranet use	8.7
More data	4.3
Further integrate with business applications and work flow	4.3
Web integration	8.7
Upgrade	4.3
	100.0

6 Views of Managers vs. Non-Managers

In our study roughly half (27 out of 57 respondents using groupware) of the sample consisted of managers with titles such as director, manager, vice-president, CIO or executive director. The others were classed as non-managers. We examined the differences in perceptions between these two groups (Table 8).

Managers tended to be more critical of how well the groupware worked; only 80% felt it worked well while 100% of the non-managers felt that the groupware worked well. Similarly the mean overall satisfaction score for non-managers was 3.393, whereas for the non-managers it was 4.036. In line with this finding in Table 8 it is evident the importance of each of the 13 applications was significantly higher for non-managers than for managers. When the uses are ranked in terms of relative importance the rank orders for managers and non-managers are very similar with rho the rank order correlation coefficient being .982.

Both groups agree on the five most important uses: (1) knowledge sharing, (2) remote access, (3) discussion and collaboration in projects, (4) workflow and (5) scheduling. And again, we see general agreement that the communication functions (functions 1-7) are ranked as more important than the other functions. There were larger differences in the rankings of the other groupware capabilities. For example, the importance for tracking customer service requests was 1.654 for managers and 3.154 for non-managers. Similarly tracking hardware and software inventory was more important for non-managers 3.269 than for managers 1.960.

As far as actual use of the groupware is concerned, for both managers and non-managers almost all used groupware for knowledge sharing (96.2% and 100.0% respectively). Eighty-nine percent of both groups used their groupware for remote access and 85% of both groups used it for discussion and collaboration in projects. Thus it seems evident that these three popular groupware packages provide support for a large majority of managerial as well as non-managerial users. Non-managers tend to use groupware for far more uses than managers do, as all the other uses were employed by non-managers far more frequently than by managers. (Table 8)

7 Sponsorship of Software

The adoption process for software is a crucial factor in most companies. In our questionnaire the process was divided into three stages: 1) learning about the product, 2) proposal to purchase the product, and 3) actual sponsorship of the product (Table 9). As far as learning is concerned, only 8.6% learned from top management (Table 9a). The most influential groups were the MIS dept (53.4%), department management (13.8%), workers in other departments (5.2%) and trade magazines (5.2%). This conforms with the findings of Rogers (1983) that most individuals do not evaluate an innovation on the basis of scientific studies but they use the recommendations of "near peers" - people who are similar but slightly more technically competent. When it came to proposals for product purchase, top management accounts for 27.2%, the MIS department 49.1% (Table 9b) and department management 9.1%. The primary sponsor of groupware (51.9%) was the MIS department (Table 9c). Second in line was top management (22.2%), third comes department management (14.8%) and 9.3% of the respondents who adopted groupware said that they themselves were the primary sponsor. There is a high correlation between who proposed it and who sponsored it (R = .562, level of significance is .001). Also a high correlation between how they learned of it and who proposed it (R = .436, level of significance is .001).

We divided the sponsors into three groups: top management, department head, and MIS department. These different sponsors may influence which software was purchased, subsequent time elapsed from proposal to purchase of the software, user group size, and hours of training. These are reported in Table 10. The brand of groupware adopted did vary by the sponsors: 63.6% of the top management sponsored Lotus Notes, in contrast 50.0% of the MIS department sponsored MS Exchange (Outlook). As far as department heads were concerned, 42.9% favored Lotus Notes and the rest were divided equally between Novell's GroupWise and MS Exchange (Outlook) (Table 10a).

Top management tends to sponsor software used by groups over 300 in size, while department heads sponsor software used by less than 30 individuals. MIS sponsors software at all levels but mostly for groups of 300 or less (Table 10b,c). When top management sponsored the software, it tended to take longer to purchase, no doubt because of the large size of the groups who would use it (Table 10d). As far as training was concerned there was no statistical difference between the average hours of training for the software suggested by top management, department heads or the MIS group (Table 10e).

Did the sponsorship relate to the evaluation of software performance? For software sponsored by top management and department heads there was unanimity that it worked well. About 11% of those using software sponsored by the MIS department felt that it didn't work well (Table 11a). Software sponsored by the MIS department also scored lower in satisfaction (Table 11b). This finding coincides with that of Downing and Clark (p. 29) who found that the six firms who did not realize higher productivity had a lower response to the question that management had sponsored the project than the 16 firms who realized higher productivity.

These two findings were contradicted by a final question: "Do you feel your group can do [their] job better using groupware?" Although MIS-sponsored software users were not as satisfied, or as pleased with the way the software worked, they were much more likely to say that it helped them to do the job better (Table 11c) than the soft-

Table 8: Importance and Use of Applications by Managers and Non-Managers

	Managers			Non-Manangers		
	Weighted Mean	Percent Using	Rank Order	Weighted Mean	Percent Using	Rank Order
Knowledge Sharing	4.30	96.2	1	4.52	100.0	1
Remote Access*	3.78	88.9	2	4.36	89.3	2
Discussion & Collaboration in Projects*	3.70	84.6	3	4.21	85.2	3
Workflow	3.41	51.9	4	3.93	64.3	4
Scheduling	3.11	55.5	5	3.89	81.5	5
Discussion Forum (Wide Groups)	3.00	53.6	6	3.83	70.4	7
Document Management	2.82	63.0	7	3.86	78.6	6
Monitoring Workflow*	2.39	29.6	8	2.85	28.6	10
Converting from Main Frame to Client Server	2.29	25.9	9	2.67	18.5	12
Building a Platform for E-Commerce*	2.08	18.5	10	2.48	22.2	13
Tracking Hardware/Software Inventory*	1.96	18.5	11	3.27	29.6	8
Launching Website*	1.92	25.9	12	2.81	25.9	11
Track Customer Service Requests*	1.65	14.8	13	3.15	51.9	9

Rank Order $\rho = .982$

*Application suggested by Gagne, 1997

ware sponsored by top management (96.3% vs. 77.8%). There may be an interaction between the sponsor, the brand of software used, and the perceived effectiveness of that software. In Table 3 we did see that MS Exchange (most frequently sponsored by top management) and Lotus Notes (department heads), had a somewhat higher satisfaction rating than Novell's GroupWise (MIS department). Unfortunately the sample was not large enough to divide these opinions by both sponsor and type of software. Finally, when the plans to extend the use of the software were explored, software sponsored by top management and by the MIS department was more likely to be considered for new adoptions and extensions (Table 11c), than software proposed by department heads.

8 Conclusions

Groupware is still clinging tightly to its roots in communication and collaboration support. The seven functionalities cited as most important and most commonly used were all the standard tools supporting groups (ranked 1-7: knowledge sharing, remote access, discussion & collaboration in projects, workflow, scheduling, discussion forum amongst widely dispersed groups, and document management). One type of business that has taken the advantage of groupware is consulting (Garcia, 1997). Since most of the employees within these companies travel extensively, the companies needed to find a way to provide these workers with the information needed to do their job. The best way to do this was through groupware technology.

The functions which branch off from that core of group support (ranked 8-13: monitoring workflow, tracking hardware/software inventory, converting from mainframe to client server, tracking customer service requests, launching a website, building a platform for electronic commerce) do not appear to be as useful to management. Many companies may handle these functions with an application built in-house (Bort, 1997). We thought that if companies had these capabilities within a groupware tool, they may use it. That does not appear to be the case. There are two potential reasons for this: 1) groupware in its current state does not effectively support those functions, or 2) management has not learned how to effectively deploy these newer IT support functions. Our respondents did not mention any of these non-communication functions in their comments on how the groupware could be improved or extended. Nor did they rate those functions as very important but not used: Table 2 shows a large degree of consistency between what functionality is important and what is used. If users had found their groupware did not support a critical function, we would also have seen differences in these ratings. We must, therefore, conclude that the problem does not lie with the groupware, but with management's view of the importance of these functions.

We see that the MIS department is a prime educator and mover in these decisions. "Push" technologies are those that the MIS department proposes to its users, rather than the users "pulling" for the new technology. Gruden has pointed out that "management is less committed to the less expensive groupware applications or features. An organization will not restructure itself for each new application the way it does around a major new system." (p. 95). It may be the case that top management still views groupware as essentially email, an older and very common technology. It is quite consistent with the idea of a "push" technology that

Table 9: How Learned About Software, Who Proposed and Who Sponsored It

a. How did you learn about it?*		b. Who proposed the product purchase**	
Top management	8.6	Top management	27.20
Department management	13.8	Department management	9.1
Workers in division	10.3	Consultant	1.9
MIS department	53.4	Worker in the division	1.8
Worker in another department	5.2	MIS department	49.1
Consultant	1.8	Worker in another department	3.6
Trade magazine	5.2	You, yourself	5.5
Other	1.7	Other	1.8
	100.0		100.0

c. Who was the primary sponsor of your groupware?	
Top management	22.2
Department management	14.8
MIS department	51.9
I was	9.3
Work in another department	1.8
	100.0

*percentages of 58 respondents for this question
**percentages of 55 respondents for this question

the users do not immediately know how to take full advantage of it. This may account for the perceived low value of the non-communication functions. MIS, as the primary sponsor, must take the initiative to better communicate how groupware can be used to leverage customer service, and to manage employees and capital. Top management still gets behind larger installations of the software, providing the necessary backing to get different departments to agree and implement a common software solution. Our data on the evaluation of groupware depending on the level of sponsorship is inconclusive. In some questions, MIS-sponsored software was viewed disfavorably, while in others it was viewed most favorably. More research is needed to determine the effect of sponsorship on perceived groupware effectiveness.

The differences between the views of managers and non-managers is important from an experimental point of view, in that researchers often use MBAs and undergraduates to conduct "satisfaction" studies of new software. Our study indicates that the ratings from these non-manager participants may be more positive than ratings provided by managers. We cannot say what contributes to the less positive ratings of managers. They may be due to a better understanding of the possible functionality of software, higher expectations toward software applications, or greater experience with other forms of groupware. But this finding does lend a note of caution to researchers who wish to claim that their software has high satisfaction ratings. That managers rate the communication functions of groupware as more important than other functions may be a result of their job definition. Managers "manage," and usually that means communicating what to do, what to change, what not to do, etc., thus the communication functions are more useful to managers in the execution of their jobs. Non-managers may be more involved in other functions such as inventory or web sites.

One last trend is offering groupware capabilities through the Internet for a specific time period (Miley, 1997). We found no indication, however, that companies were shying away from groupware because of the cost. No one mentioned cost as a significant concern, either in why they were not using groupware, nor in how it could be improved. Our sample included mostly large companies, however, so perhaps this may be of more concern for small companies.

Groupware mostly helps companies communicate better within themselves. It still needs to expand its uses in areas other than group communication. Finally, it may also help companies communicate better between themselves. When four of the big six accounting firms contemplated mergers in 1997 (Price Waterhouse and Coopers and KPGM Peat Marwick and Ernst & Young) these firms were among the heaviest users of workgroup software. It was noted that companies using groupware generally have an easier time merging because their corporate cultures are already committed to information sharing (Cole-Gomolski 1997). With the recent mergers among the high-tech companies, we may find groupware playing a larger part in determining what companies merge, and how effective those mergers are.

Acknowledgement

The authors wish to thank Monica Miera of the Microelectronics Group of Lucent Technologies for her research and consultation help with this paper.

Table 10: Relation of Sponsorship to Purchase Conditions

	percent of Top Management	percent of Department Head	percent of MIS Department
a. Type of Software Adopted			
Lotus Notes	63.6	42.9	35.7
Novell's GroupWise	18.2	28.6	14.3
MS Exchange (Outlook)	18.2	28.5	50.0
	100.0	100.0	100.0
b. Average size of group			
1 - 30 people	-	66.7	40.0
31 - 300	33.3	33.3	35
More than 300	66.7	-	25
	100.0	100.0	100.0
c. Total size of group			
1 - 30 people	-	14.3	15.4
31 - 300	10%	49.2	34.6
More than 300	90	42.8	50
	100.0	100.0	100.0
d. Time needed for purchase			
3 months or less	-	25.0	36.8
4 months or more	100.0	75.0	63.2
	100.0	100.0	100.0
e. Hours of training			
2 hours or less	42.9	33.3	47.1
3 hours or more	57.1	66.7	52.9
	100.0	100.0	100.0

References

[1] Ahuja, S. R., Ensor, J. R., Koszarek, J. L., & Pack, M. (1989) Supporting Multi-Phase Groupware Over Long Distances. *Proceedings of IEEE Global Telecommunications Conference.*

[2] Bort, Julie (1997) Groupware on the Net. *VARbusiness*, 13, 17, p. 75-6.

[3] Cole-Gomolski, Barb (1997) Groupware Put To Test. *Computerworld*, 31, 43, p. 14.

[4] *Computerworld.* (1994) The Premier 100. September 19, p. 49-53.

[5] *Computerworld*, (1995) October 9, p. 46-52.

[6] DeSanctis, G. R. & Gallupe, B. (1987) A Foundation for the Study of Group Decision Support Systems. *Management Science*, 33, 5, p. 589-609.

[7] Downing, Charles E. & Clark, Andrew (1999) Groupware in Practice: Expected and Realized Benefits. *Information Systems Management*, 16, 2, p. 25-31.

[8] Ensor, J. R. (1990) How Can We Make Groupware Practical? *Proceedings of ACM, Converence on Human Factors in Computing Systems*, p. 87-9.

[9] Gagne, Cathleen A. (1997) Grapplin' With Groupware? *Computerworld*, Sept. 30, p. 83.

[10] Garcia, Mary Ryan (1997) Knowledge Central. *Information Week*, 649, pp. 252-6.

[11] Grudin, Jonathan (1994) Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM*, 37, 1, p. 92-105.

[12] Hirokawa, R. Y. & Rost, K. M. (1992) Effective Group Decision Making in Organizations. *Management Communication Quarterly*, 5, 3 p. 267-288.

[13] Lavilla, Stacy, & Walker, Christy (1997) Oracle Tunes Apps. *PC Week*, 14, 40, p. 10.

[14] Likert, Renses (1970) A Technique for the Measurement of Attitudes in *Attitude Measurement*, ed. Gene F. Summers, Chicago, IL: Rand McNally, p. 149-158.

[15] Maurer, Todd J. & Pierce, Heather R. (1998) A Comparison of Likert Scale and Traditional Measures of Self Efficacy. *Journal of Applied Psychology*, 83, 2 p. 324-332.

[16] Miley, Michael (1997) Groupware Technology Moves to the Internet. *MacWeek*, 10, 1, p. 73-4.

Table 11: Evaluation of Software by Sponsor

	Percent of Top Management	Percent of Department Head	Percent of MIS Department
a. Works Well			
Yes	100.0	100.0	88.9
No, unknown	-	-	11.1
	100.0	100.0	100.0
b. Satisfaction			
Weighted mean	4.2	4.0	3.8
c. Helps to do the job better			
Yes	77.8	85.7	96.3
No, unknown	22.2	14.3	3.7
	100.0	100.0	100.0
d. Plans to extend			
Will extend	71.4	50.0	63.6
Won't extend	28.6	50.0	36.4
	100.0	100.0	100.0

[17] Nunamaker, J. F. Jr. & Vogel, D. R. (1991) A Comparison of Laboratory and Field Research in the Study of Electronic Meeting Systems. *Journal of Management Information Systems*, 7, 2, p. 107-135.

[18] Orlikowski, Wanda J. & Hofnan, J. Debra (1997) An Improvisational Model for Change Management; the Case of Groupware Technologies. *Sloan Management Review*, 38, 2, p. 11-21.

[19] Pervan, Graham P. (1998) A Review of Research in Group Support Systems: Leaders, Approaches and Directions. *Decision Support Systems*, 23, 2, p. 149-159.

[20] Radosevich, Lynda. (1997) Novell Gears Apps for Web. *InfoWorld* 19, p. 14.

[21] Rogers, Everett M. (1983) *Diffusion of Innovations*, 3rd ed., New York: Free Press.

[22] Thomas, Susan L. (1997) Schedule Standard Penciled in. *LAN Times*, 14, 22, p. 36.

[23] Wheeler, Bradley C. & Valacich, Joseph S. (1996) Facilitation, GSS and Training as Sources of Process Restrictireness and Guidance for Structured Group Decision Making: An Empirical Assessment. *Information Systems Research*, 7, 4, p. 429-448.

APPENDIX: GROUPWARE QUESTIONNAIRE

(blank response lines omitted)

1.What is the function of your department?
a. How many people are in your department?
b. What is your title?

2. Do you currently use any form of groupware? (ie. Lotus Notes, Novell's Group Wise, or Microsoft's Exchange) in your department?
Which one?
If No, do any departments in your company use groupware?

3. Do you use a Web based package?
If Yes, which one?
Please list the departments using groupware and answer all questions in terms of this department.
Which department?
Which groupware?

If No, to questions 2 and 3 please go to questions 21-24.

4.How did you learn about your groupware?
a. Top management
b. Department Management
c. Worker in your division
d. MIS Department
e. Worker in another company or division
f. Software salesperson
g. Other (Specify)

5.Who first proposed purchase of this groupware?
a. Top management
b. Department Management
c. Worker in your division
d. MIS Department
e. Worker in another company or division
f. Software salesperson
g. You, yourself did
h. Other (Specify)

6.Who was the primary sponsor of this groupware?
a. Top management
b. Department Management
c. Worker in your division
d. MIS Department
e. Worker in another company or division
f. Software salesperson
g. You, yourself did
h. other (specify)

7.Please answer some questions about your software purchase.
a.When did you adopt it?
b.How long did it take from the time this was first suggested to purchase?
c.How long have you been using it?

8.Please describe how your groupware has worked out.
a.Does it work well?
b.How big was the average size group using this product (# people)?
c.How many people altogether do you estimate use it now?

9.What were you primary reasons for adopting this software?

10.Do you feel that the use of this groupware has increased your groups' productivity?
a.(If no) Why not?
b.(If yes) In what way?
c.(If yes) Can you estimate how much money it saves per month?
d.(If yes) Can you estimate how much time it saves per month?
e.(If yes) How many months do you think it will take to payback your investments in software and training on this particular groupware?

11. Was your purchase of this groupware strongly influenced by your need to use the Worldwide Web?
If Yes, in what way?

12. How many people are connected (# of users) in your groupware use?

13. How satisfied are you with this groupware?
From 1 - Not Satisfied
to 5 - Very Satisfied
1 2 3 4 5

14. Below is a list of popular applications for groupware. Please rate their importance to you or your company on the following scale by circling a number from 1 through 5, where 1 indicates that the reason is unimportant and 5 indicates that the reason is very important.
a. For remote access and functionality across many locations
1 2 3 4 5
b. For knowledge sharing
1 2 3 4 5
c. For document management
1 2 3 4 5
d. For workflow
1 2 3 4 5
e. For discussions and collaborative work within project teams
1 2 3 4 5
f. For discussion forums amongst a wide group
1 2 3 4 5
g. To convert from a mainframe to a client server system
1 2 3 4 5
h. To monitor workflow
1 2 3 4 5
i. To build a platform for electronic commerce
1 2 3 4 5
j. To track customer service requests and purchase orders
1 2 3 4 5
k. To track hardware and software inventory
1 2 3 4 5
l. To launch a web site
1 2 3 4 5
m. For scheduling
1 2 3 4 5

15. For each of these applications, do you use your groupware for this function?
a. For remote access and functionality across many locations (Yes No Don't know)
b. For knowledge sharing (Yes No Don't know)
c. For document management (Yes No Don't know)
d. For workflow (Yes No Don't know)
e. For discussions and collaborative work within project teams (Yes No Don't know)
f. For discussion forums amongst a wide group (Yes No Don't know)
g. To convert from a mainframe to a client server system (Yes No Don't know)
h. To monitor workflow (Yes No Don't know)
i. To build a platform for electronic commerce (Yes No Don't know) j. To track customer service requests and purchase orders (Yes No Don't know) k. To track hardware and software inventory (Yes No Don't know)
l. To launch a web site (Yes No Don't know)
m. For scheduling (Yes No Don't know)

16. Did you have training for your users of groupware?
(If Yes), approximately how many hours of training did you offer per user?
(If Yes) How many users did you train?
(If Yes), was the training in house or did you use outside trainers or both?
(If you used a combination of in-house and outside, which was most effective? Why?

17. What functions of your groupware were useful? Why?

18. How could the groupware you use be improved?

19.Do you feel your group can do job better using groupware? Yes, No, or Don't Know?

20.How would you like to improve the functionality of your groupware?
For All Respondents:
21. Do you think you will adopt or extend your use of groupware in the next year?
(If yes) How?
What software would you need ?

22. Did you consider using groupware, and then not accept it?
(If Yes) why did you decide not to adopt it?
(If no) why not?

23. If you would like a copy of this report, please give us your name and address.

PLEASE PUT THIS QUESTIONNAIRE IN THE ENCLOSED, STAMPED ENVELOPE AND RETURN IT TO US. THANK YOU FOR YOUR PARTICIPATION.

An Empirical Study to Measure the Diffusion of GroupSystems in Organizations

Morgan M. Shepherd
University of Colorado,Colorado Springs, USA,
Phone: (719) 262-3641, Fax: (719) 262-3494,
E-mail: mshepher@mail.uccs.edu

Keywords: Group Support Systems, Diffusion, Work Groups, Facilitator

Edited by: Gary Klein
Received: June 11, 1999 **Revised:** December 20, 1999 **Accepted:** December 22, 1999

This research was concerned with determining the major factors that affect the diffusion of GSS (specifically GroupSystems) in organizations. The variables that had the most significant effects were the size of the work groups within the organization, the hourly rate charged to use the technology, the amount of money initially spent on the technology and the role of the facilitator. But the existing models have some gaps, and some suggestions for improving diffusion research are discussed.

1 Introduction

"Do more with less" has become the latest "buzz phrase" in corporate America. Organizations are right-sizing, middle management levels are being removed from the organizational structure, and employees are being challenged to increase their output in the face of these cutbacks (Luthans, 1995; Markus, 1994). Emphasis has been placed on group activities (Applegate, 1991) as organizations move toward greater use of project teams or work teams to complete tasks and solve problems. There is a shift away from past reliance on individuals and toward increasing reliance on interfunctional work groups to solve complex problems (Luthans, 1995).

Many organizations have been turning to a new type of technology, called Group Support Systems (GSS), to increase the productivity of these workgroups (Mills, 1995; Lyon, 1995). Numerous articles have appeared concerning the need for organizations to utilize information technology (IT) to gain/maintain a competitive advantage (Mills, 1995; Porter, 1990; Dennis, Nunamaker, and Paranka, 1991). According to these researchers, the key to gaining/maintaining a competitive advantage is to identify appropriate technologies and quickly diffuse them to end-users, i.e. to work groups. While there has been a lot of diffusion research on individual user technologies, there has been very little research on how to diffuse these group-enabling technologies (GSS) throughout organizations. The purpose of this research is to identify the major critical factors that need to be addressed to more fully diffuse GSS within organizations.

There are several distinct differences between individual technologies (i.e. word processor or spreadsheet programs) and GSS that limit the generalizability of the existing diffusion research. Individual technologies are easier to obtain due to their lower cost and the shorter decision process involved with the purchasing decision (basically just one person deciding). The technical support is minimal, the interface can be modified to each user's liking, and it is fairly easy to view the technology prior to purchasing it. The technology is fairly easy to use, help is readily available, and the critical mass needed is one person. Conversely GSS are more costly, have greater technical support needs (networking support), are more difficult to use, and require a certain critical mass of users in order for the organization to realize any gains. And as GSS are by nature group-enabling tools, the decision process to purchase usually involves many people.

1.1 Reasons for specifically researching GroupSystems

Prior research demonstrates that GroupSystems increases the productivity and effectiveness of work groups and project teams. Voted "best of breed" in 1994 by PC Magazine, GroupSystems has a well-documented track record for saving time (over 50% in meeting times and 90% in project time, Martz, Vogel, and Nunamaker, 1992) and money (Post, 1992). At the time this research was conducted, GroupSystems was used in over 500 different organizations, including government, universities, and Fortune 100 corporations, some of which are using GroupSystems in more than one site. The fact that it is the 'best of breed' and is currently used in hundreds of sites made GroupSystems a primary candidate to research.

1.2 Research Question

GroupSystems has not yet achieved a critical mass (Briggs et.al., 1998). Given its having been approved by so many

adopting organizations and that there is wide acknowledgment that IT is necessary for organizations to gain/maintain competitive advantage, why hasn't GroupSystems been more completely diffused among those who have access to it? This is not a problem of *implementation*, which is primarily concerned with the introduction of a technology into an organization. All of the organizations researched had already approved the business case for adopting GroupSystems and GroupSystems had been installed and implemented by at least one department in each of those organizations. This leads to the research question; what are the factors that influence the diffusion of GroupSystems in organizations?

2 Prior Diffusion Research

Much of the diffusion research stems from Rogers's work on the diffusion of innovations (1983). He has defined five major characteristics along which innovations differ: 1) relative advantage, 2) compatibility, 3) complexity, 4) trialability, and 5) observability. Most of Rogers's work did not involve technology. Nonetheless, subsequent technology oriented researchers have built on his work and identified the following factors that affect diffusion: characteristics of the technology, the perceived advantages of the technology, the nature of the communication channels that exist and are utilized by the organization, the cultural aspects of the organization, the organizational structure, the individual characteristics of the employees, the business environment and the technical support capabilities of the organization (Straub, 1994; Amoroso and Brancheau, 1990; Brancheau and Wetherbe, 1990; Lazarsfeld and Menzel, 1963; Ryan and Gross, 1943). However, most of this research deals with individual technologies, those technologies that support a single user at a time. This includes Davis' TAM model (Davis, 1989; Davis et.al. 1989). Although the TAM model is the most widely validated technology diffusion model, it was developed by examining the diffusion of a single user technology called WriteOne.

2.1 Diffusion research deficiencies

One of the challenges for GSS diffusion researchers is that all technologies, or innovations are not alike (Prescott and Conger, 1995). To further complicate matters the diffusion research discipline lacks a standard set of terminology. There is also a lack of consensus concerning the boundaries encompassed by the term diffusion. In a review of the diffusion literature, Prescott and Conger (1995) found papers that used the term diffusion inconsistently, found papers in which the constructs were not accurately operationalized, found papers in which the innovation being researched was not defined, and found papers that did not clearly define what specific stage of the diffusion process was being researched.

In terms of applicability to GSS, the current models of diffusion research deal with individual technologies. From the diffusion research perspective there is a dearth of research concerning this aspect of group technologies. And from the GSS perspective, there has been very little research on the diffusion of group support systems in organizations. Section 2.2 discusses the prior GSS diffusion research.

2.2 Prior GSS diffusion research

Although little has been researched on the diffusion of GSS, GSS researchers have identified the following factors that help in the implementation of GSS. The need for a strong internal champion, finding a willing adopting group or department as a "guinea pig", making sure the initial uses of the technology have an extremely high chance of being successful, managing the communication of the success(es) of the initial meeting(s) throughout the organization, making sure the technology fits with the organizational culture, and the role of the facilitator have all been identified as being important in the successful implementation of GSS within organizations (Alexander et al., 1992; George et.al., 1992; Beath, 1991; McKenny and Mason, 1995; Clawson, Bostrom, and Anson, 1993; Dennis et al., 1988; Nunamaker et al., 1987, 1988, 1989, 1991).

2.3 Building on the prior diffusion research

The goal of this project was to build upon the existing diffusion research and the existing GSS implementation research by examining some of the variables specific to group support technologies. This earlier GSS research was used as a starting point to determine which factors, or variables would be studied in this project. In addition, interviews with some of the existing sites where GroupSystems was well diffused (Chevron in CA, the Department of Defense in D.C., and the University of Arizona) and where GroupSystems was phasing out (IBM in Gaithersburg, MD) were used as a sanity check for the variables.

This research was meant to be exploratory and is not to be considered exhaustive by any means. However, the variables that were used are considered to be a logical starting point for this type of research.

3 Hypotheses

The first hypothesis came from the first leg of Rogers's framework, which states that the diffusion process is helped when potential adopters have an opportunity to see the relative advantage of an innovation. To see the advantage that GroupSystems can provide, access to the technology would need to be readily available to the potential adopters. This hypothesis was also given support as a result of the interviews. IBM found that use of GroupSystems dropped off significantly when groups were charged for the use of the technology.

H1: GS will diffuse more in organizations that charge less for the use of the GS facility than in organizations that charge more for the use of the GS facility.

The next two hypotheses came from the interviews and the diffusion literature. As GroupSystems is a group-enabling software tool it makes sense that organizations that emphasize group work should have more success diffusing the technology than groups that do not emphasize group work or use project teams. Chevron found this to be true in their organization.

H2a: GS will diffuse more in organizations that use project teams or work groups to accomplish most of the work than in organizations that do not use project teams or work groups to accomplish most of the work.

H2b: GS will diffuse more in organizations with larger work groups than in organizations with smaller work groups.

Hypothesis 3a and 3b follow from the GSS implementation literature concerning the effects of the facilitator role, and were strongly agreed with by all interviewers.

H3a: GS diffusion is correlated to the number of facilitators within the organization.

H3b: GS diffusion is correlated to the number of trained, full-time facilitators within the organization.

Hypothesis 4 also follows from the GSS implementation literature concerning the importance of a strong internal champion, and was also strongly supported by the interviews.

H4: GS diffusion is inversely correlated to the number of levels of command between the internal champion and the person with sign-off capability.

Hypothesis 5 did not come directly from the GSS or diffusion literature, but primarily from the interviews. In today's business environment most employees and departments are primarily evaluated on a short-term basis, quarterly or at most yearly and new technologies are rated on cost per person to the organization. It is logical to assume that a high-priced technology will be difficult to implement ("That's a lot of money to spend. Are we sure about the rate of return?") It is also logical to assume that a high-priced technology will be given more chance to diffuse than a low priced technology. ("We've spent too much on that technology to give up on it just yet. Let's give it another year before we decide.") This lead to the fifth hypothesis:

H5: The degree of diffusion will be correlated with the total amount spent on the GroupSystems facility.

3.1 A Brief Explanation of the Variables that were Analyzed

The dependent variable was the degree of diffusion of GroupSystems. This variable was measured by dividing the number of employees at the particular site into the number of employees who were GroupSystems users. This provided a percentage, which was referred to as the degree of diffusion for that particular organization. This permitted the results to be applied across all types of organizations, even though the organizations studied were of differing sizes.

In diffusion research the measurement of this variable is difficult. The biggest problem is to accurately determine who qualifies as a user. For the purpose of this study, a user was defined as a repeat user of GroupSystems. Although it could be argued that this might artificially reduce the rate-of-diffusion numbers, it was consistently done with all of the organizations. This method eliminates those who have tried GroupSystems but did not like the technology. It was felt that these people should not be included as users. Conversely, it also eliminates those who tried GroupSystems, liked the technology, but have not had the need or opportunity to use it a second time.

Based on the prior diffusion research, GSS research and the interviews, the following independent variables, also shown in Table 1, were analyzed: the number of available facilitators, the number of full time facilitators, the amount of money that is charged to use the GroupSystems facility, the percentage of work that is accomplished by using project teams or groups, the average size of the workgroups, the amount of money that was spent on the GroupSystems facility, and the position of the internal champion within the organization.

Facilitation refers to the availability, and the quality (trained vs untrained) of the GroupSystems facilitator(s) in the organization. In following Rogers work, facilitators are the ones who can best demonstrate to potential users the benefits and ease of use of GroupSystems. Competent facilitators can also demonstrate the superiority of a GroupSystems meeting versus a non-GroupSystems supported meeting. You would probably be more convinced to adopt a word processor from seeing someone use it well, than from seeing someone who struggled to get it to do what they wanted. Therefore it is important to know the number of facilitators in the organization, defined in this research as anyone who knows how to use the technology. It is also important to know the number of full-time facilitators, defined as facilitators who went through the formal GroupSystems training program and whose full time job is facilitation.

Access to the facility will be easier if there are fewer barriers in place. The amount of money that is charged to use the facility is one of the main barriers to use in GroupSystems organizations. Measuring this variable should indicate what affect charging for the use of the GroupSystems facility has on the diffusion of GroupSystems.

Table 1: The Independent Variables

Variable name	What the variable accounted for:
Number of Facilitators	The number of facilitators in the organization. A facilitator was defined as someone who had run the GroupSystems software
Number of full time Facilitators	A full time facilitator is a facilitator who has received formal GroupSystems training, and who's sole responsibility is facilitation related work for forty hours a week
Money spent on the GS facility	The total amount of money spent on the GS facility
Hourly charge for the use of the GS facility	The internal hourly charge (if any) for the use of the room, in dollars/hour
Average department size	The average size (in number of people) of departments
Average work group size	The number of people in the average work group
Percentage of work accomplished in groups	The overall percentage of work that is accomplished via work groups or project teams
Percentage of repeat GS users	The percentage of repeat GS participants

The amount of work accomplished with groups or project teams provides an indication of how well GroupSystems fits with the organization. This follows closely from Rogers's second characteristic, compatibility. Innovations that are more compatible with an organization should have higher rates of diffusion than those innovations that are less compatible. Measuring the amount of group work seems logical for this research, as GroupSystems is a group-enabling technology. GroupSystems works with groups of all sizes, but provides better gains for large groups. This is also a compatibility issue in that GroupSystems fits better in organizations that use larger work groups. Therefore the average size of the work group was included in this research.

The amount of money spent on the GroupSystems facility was believed to have an effect on the degree of diffusion. In today's business environment most employees and departments are primarily evaluated on a short-term basis. It is logical to assume that a high-priced technology may be given more chance to diffuse than a low priced technology.

The internal champion also plays a role in the diffusion process. Although much of the internal champion information comes from the implementation literature, the role of the internal champion in relation to upper management was observed in the study to see its effect on the degree of diffusion. It was anticipated that if the internal champion were higher in the organization than the person with authority to purchase the technology, the technology would diffuse more readily than in an organization where the internal champion was lower in the organizational hierarchy than the person with purchase authority.

There are many other factors that could be included, but each of the included variables represents a major area from one of the mentioned research streams. The research model for this research came from these variables, and is shown in Figure 1.

4 The Research Approach

A survey methodology was used to collect data from a wide range of respondents, who represented a random sample of the 500 organizations that had implemented GroupSystems by October 15, 1994. The survey was administered to either the internal champion of GroupSystems or to a facilitator in each of the organizations. These people were chosen because of their overall knowledge of the organizational history of GroupSystems. One hundred surveys were sent out, with a response rate of 45%. The sample included government, commercial, and research sites. The average site had approximately 1500 potential users, and had spent approximately $110,000 on the GroupSystems facility. Seventy-eight percent of the surveys came from either government or commercial sites, and twenty-two percent were from research cites. Descriptive statistics of the research variables from the data set are in Table 2.

A regression analysis was used to interpret the data. Due to the exploratory nature of this research, a p-value of .10 was used to increase the ability of the model to detect differences (Jarvenpaa, Rao, and Huber, 1988). Stepwise, forward, backward, and standard regression analysis techniques were used, with the PIN and POUT values set at .10 and .15 respectively. The regression equation shows which variables were significant in their effect on the degree of diffusion.

4.1 Data Analysis

The data were run through four types of regression analysis; forward, backward, stepwise, and a straight regression. Three of the four models had identical variables, with the straight regression model differing by the inclusion of one additional significant variable. As the stepwise model is the most common method of running regressions, the output from that model was used. The variables that were included in the regression are shown in the table below. There was

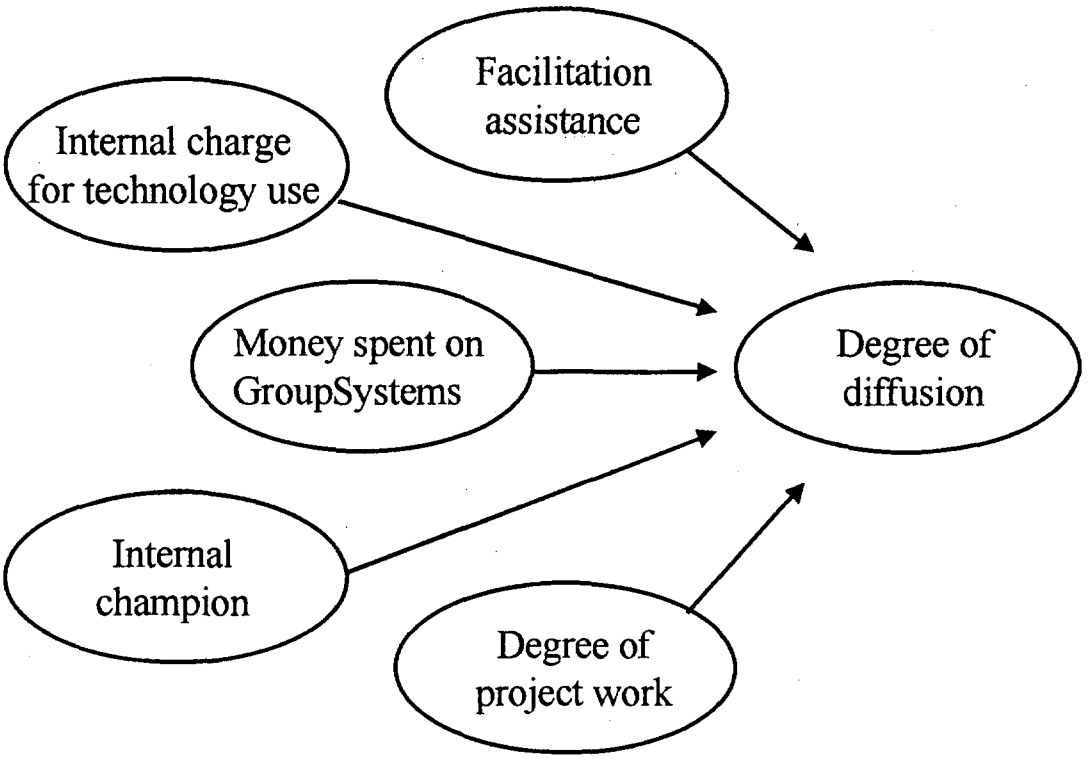


Figure 1: GroupSystem Diffusion Research Model

no significant correlation between any of the variables. The regression analysis indicated that 58% of the model can be accounted for by these variables, and is significant at the p <.001 level. The results of the regression are in table 3 and the regression equation is given by:

Degree of diffusion = - .036
+ .013 (Size of the work groups)
+ .086 (Number of full-time facilitators)
- .001 (The hourly charge)
- .001 (% of work done by work groups)
- .003 (The number of facilitators)
+ .001 (Money spent on the technology).

$F = 9.301$, $p < .001$, $R^2 = .806$, Adjusted $R^2 = .580$

5 Discussion of Results

The regression equation shows those variables that had a significant effect on the degree of diffusion of GroupSystems. The number of levels of command between the internal champion and the person with sign-off capability was the only independent variable that did not have an effect on the degree of diffusion of GroupSystems. Six of the seven hypotheses were supported, but not all in the predicted direction. As the internal champion variable was not in the regression equation, no support was found for H4. Support was found for H1, H2b, H3b and H5. Support was also found for H2a and H3a but all in the opposite direction. The research model is shown in Figure 2.

It seems intuitive that decreasing the amount charged to use the GS facility (H1), increasing the size of the work groups (H2b), increasing the number of full-time facilitators (H3b), and the total amount of money spent on the GS facility (H5) would have a positive effect on the rate of diffusion of GroupSystems. What is not intuitive is the fact that increasing the number of facilitators (full- time, formally trained & part-time, not trained) had a negative effect on the degree of diffusion. Follow-up interviews with a few of the sites only led to partial solutions. It may be that the first impression some groups had of GroupSystems was at the hands of a non-trained facilitator. It is possible that the session did not run smoothly, did not run at all, or that some data may have been lost due to the inexperience of the facilitator. This could have left a negative impression with the group and might have hindered the diffusion process.

Given that GSS are designed to support groups, it also seems counter-intuitive that increasing the amount of work performed by work groups would have a negative effect on the degree of diffusion. Follow-up interviews with some of the sites did provide some insight into this. It may be possi-

Table 2: Descriptive Statistics

Construct	Mean	Min.	Max.
Percentage of repeat GS users	20.40%	0	100%
Number of potential GS users at the site	1495	3	10000
Number of GS facilities at the site	1.4	0	4
Percentage of time the GS facility is in use	32.30%	0	95%
Average department size (number of people)	30.81	3	75
Percentage of work accomplished via work groups	49.62%	1	100
Average work group size (number of people)	8.78	0	20
Number of facilitators at the site	5.49	0	50
Number of trained facilitators	1.62	0	5
Amount spent on the GS facility	$112k	0	$560k
Hourly internal charge to use GS	$21.35	0	$200
Vertical layers separating the internal champion from the person with approval authority for additional GS	1.57	0	9

Table 3: Regression results

	B	T	Sig. T
Intercept	-0.0357	-1.5290	0.1366
Size of work group	0.0128	6.233	0.0000
Number of full-time facilitators	0.0860	4.510	0.0001
Hourly charge	0.0007	-3.4880	0.0015
Percentage of work done by work groups	-0.0013	-4.050	0.0003
Number of facilitators	-0.0034	-2.703	0.0112
Amount of money spent on the technology	0.0002	2.070	0.0471

ble that some minimum amount of group work is required to ensure adequate diffusion. It was rationalized that as the amount of group work increases too much demand may be put on the GS facility. (The average number of GS facilities was approximately 1.5, while the average number of potential users was approximately 1500.) Groups who depend on having the technology available may become disheartened when they do not have ready access to it. Imagine if you could only use your current word processor software 30% of the time you needed it, but could always use the DOS version of that software. You would probably not use your current word processor, even though it is far superior.

6 Summary and Conclusions

To improve the diffusion of GroupSystems within an organization this research found support for providing trained facilitation support, reducing or eliminating the hourly charge to use the GroupSystems facility, increasing the size of the work groups, and investing sufficient money into the GroupSystems facility. It was also found that increasing the number of untrained facilitators negatively affects the diffusion of GroupSystems, and that having too much group work may also hurt the diffusion of GroupSystems. These findings all seem to follow the general research base provided by prior diffusion and GSS researchers. Allowing the potential adopters to see GroupSystems when they want to, and having them observe GroupSystems in a favorable light seems to improve the diffusion of the technology. Providing poor assistance and putting up barriers to use seems to negatively impact the diffusion of GroupSystems. While this seems obvious, barriers to GroupSystems use are being implemented in many of the adopting organizations. Those organizations may be viewing the GroupSystems facility as a cost center and are trying to recoup the cost of the facility. It may be better to view GroupSystems as the productivity tool that it is.

Follow-up interviews indicated that some of the sites are moving to the distributed mode for using GroupSystems. This would alleviate the need to meet in the GroupSystems facility and should reduce the demand for the facility. However, this may create a steeper learning curve for potential adopters as they may have to handle the learning curve primarily on their own.

6.1 Limitations of this research

As with all exploratory research, there are some areas where this project can be improved by future researchers. The first issue is how to determine when GroupSystems (or

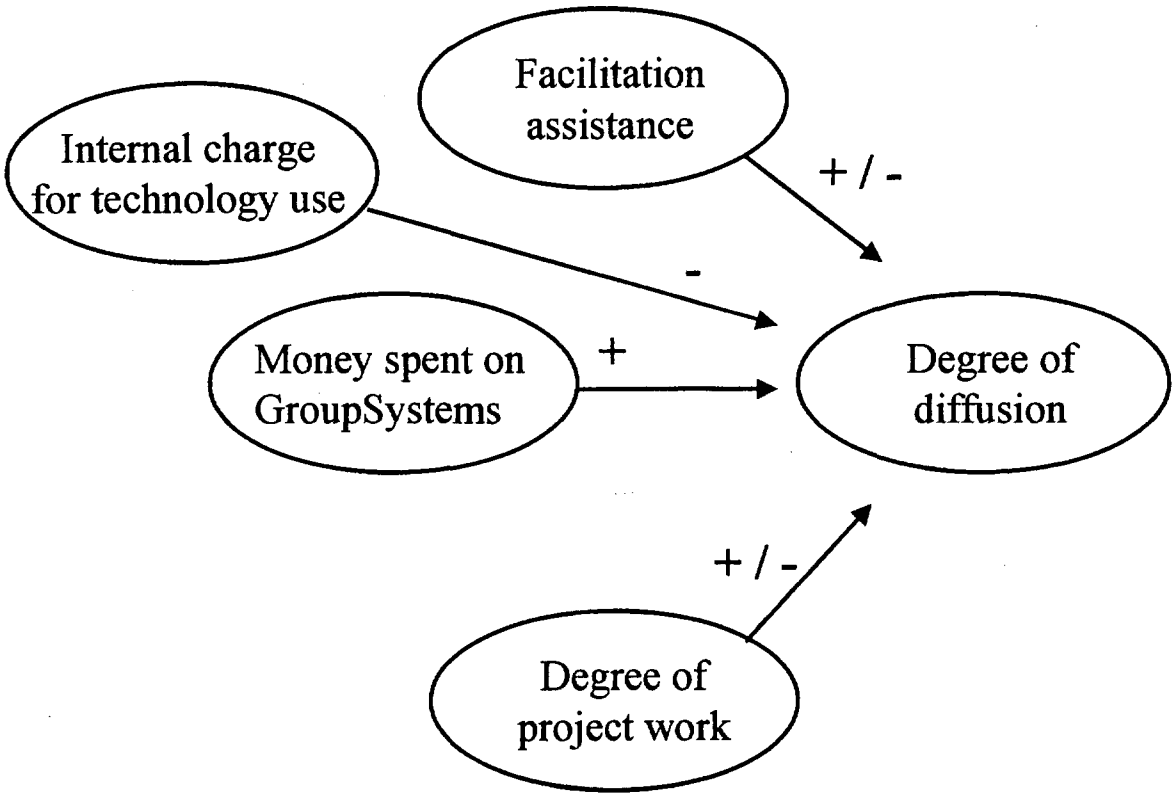


Figure 2: Final GroupSystem Diffusion Research Model

any technology) has been 'diffused' from one area of an organization to another. For this research project, a GroupSystems user was defined as someone who had used the technology at least twice. There is nothing magical about having used it twice; future researchers may want to raise that number. It may be better to rate users based on the percentage of the time they used GroupSystems when the opportunity arose. Someone who only had need of the technology twice in one year but used it both times may be a better data point than someone who had need of the technology 50 times in one year but only used it 5 times.

It would be desirable to watch the diffusion process when the technology is newly implemented. This was not possible with this research project. This allows the researcher to better control for the variables.

It is also recommended that future researchers show the spread of the technology by mapping the physical location of the users. Floor plans should be readily available from the personnel planning, or facilities department of most organizations. This will permit diffusion to be measured in a number of ways, both by use and by physical location of the technology. This could help organizations better understand how the technology is being diffused. Rogers mentions communization channels and how they can aid in the diffusion process. Graphically plotting the diffusion and then checking to see what events occurred as new areas or people use the technology could help researchers better understand how the different factors affect the diffusion process. It may be that one areas' 'internal champion' got transferred, and brought news of the technology with her. It may be that an excellent facilitator moved to another facility and helped the diffusion process. In any event, plotting the diffusion process should help alert researchers to what is happening within the organizations.

References

[1] Alexander, M., Elan, J., and Wasala, C., "Multiple Theoretical Perspectives for Studying the Assimilation of Emerging Information Technologies," Proceedings of the 25th Hawaii International Conference on Social Systems, IEEE Computer Society Press, Kauai, Hawaii, vol. IV, (January, 1992), p. 428-436.

[2] Amoroso, D.L., and Brancheau, J.C., "Using the Expansion-control Framework for Measuring Management Action in Support of Emerging Technologies," Proceedings of the 23rd Annual Hawaii International Conference on Systems Sciences, IEEE Computer Society Press, Kona, Hawaii, Vol. IV, (January 1990), p. 428-436.

[3] Applegate, L.M., "Technology Support for Cooperative Work: A Framework for Studying Introduction and Assimilation in Organizations," *Journal of Organizational Computing*, 1, (1991), p. 11-39.

[4] Beath, C.M., "Supporting the Information Technology Champion," *MIS Quarterly*, 15, 3, (1991), p. 355-372.

[5] Briggs, R.O., Adkins, M., Mittleman, D, Kruse, J., Miller, S. and Nunamaker, J.F., "A Technology Transition Model Derived from Field Investigation of GSS Use Aboard the U.S.S. CORONADO," *Journal of Management Information Systems*, 15, 3, Winter 1998-1999, p. 151-196.

[6] Brancheau, J.C. and Wetherbe, J.C., "Testing and Extending Innovation Diffusion Theory in the Context of End-user Computing," *Information Systems Research*, 1, 2, (1990), p. 115-143.

[7] Clawson, V.K., Bostrom, R.P., and Anson, R., "The Role of the Facilitator in Computer Supported Meetings," *Small Group Research*, 24, 4, (1993), p. 547-565.

[8] Davis, F.D. "A Technology Acceptance Model for Empirically Testing New End-User Information Systems: Theory and Results," Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 1986.

[9] Davis, F.D., Bagozzi, R.P. and Warshaw, P.R., "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science*, 35, 8, 1989, p. 982-1003.

[10] Dennis, A. R., George, J.F., Jessup. L. M., Nunamaker, J. F., Jr. and Vogel, D. R., "Information Technology to Support Electronic Meetings," *MIS Quarterly*, 6, (1988), p. 591-624.

[11] Dennis, A.R., Nunamaker, J.F., and Paranka, D., "Supporting the Search for Competitive Advantage," *Journal of Management Information Systems*, 8, 1, (Summer 1991), p. 5-36.

[12] George, J., Nunamaker, J.F., and Valacich, J.S., "Electronic Meeting Systems as Innovation: A Study of the Innovation Process," *Information and Management*, 22, 3, (1992), p. 187-195.

[13] Jarvenpaa, S.L., Rao, V.S., and Huber, G.P., "Computer Support for Meetings of Groups Working on Unstructured Problems: A Field Experiment," *MIS Quarterly*, 12, 4, (1988), p. 645-665.

[14] Lazarsfeld, P.F., and Menzel, H., "Mass Media and Personal Influence," in Wilbur Schramm (ed.), *The Science of Human Communication*, New York, Basic Books, 1963.

[15] Luthans, F., *Organizational Behavior*, McGraw Hill, New York, 1995.

[16] Lyon, K.W., "How to Sell GroupWare to Your Organization," *GroupWare '95 Proceedings*, (1995), p. 463-490.

[17] Markus, M.L., "Electronic Mail as the Medium of Managerial Choice," *Organizational Science*, 5, 4, (1994), p. 502-527.

[18] Martz, W.B., Vogel D., Nunamaker, J.F., "Electronic Meeting Systems: Results from the Field," *Decision Support Systems*, 8, (1992), p.141-158.

[19] McKenney, J.L., and Mason, R.L., *Waves of Change: Business Evolution Through Information Technology*, Harvard Business School Press: Boston, MA, 1995.

[20] Mills, S., "Unleashing the Power of Teamwork ... It's About Competing not Computing," *GroupWare '95 Proceedings*, (1995), p. 83-92.

[21] Nunamaker, J.F., Jr., Applegate, L. M. And Konsynski, B. R., "Facilitating Group Creativity with GDSS", *Journal of Management Information Systems*, 3, (1987), p. 5-19.

[22] Nunamaker, J.F., Jr., Applegate, L. M. And Konsynski, B. R., "Computer-aided Deliberation: Model Management and Group Decision Support", *Journal of Operations Research*, (November-December, 1988), p. 826-848.

[23] Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., and George, J.F., "Electronic Meeting Systems to Support Group Work," *Communications of the ACM*, 34, 7, (1991), p. 40-61.

[24] Nunamaker, J.F., Jr., Vogel, D., Heminger, A., Martz, B. Grohowski, R. And McGoff, C., "Group Support Systems in Practice: Experience at IBM", *Decision Support Systems*, 5, 2, (1989), p. 183-196.

[25] Porter, M., *The Competitive Advantage of Nations*, The Free Press, Macmillan, Inc., New York, 1990.

[26] Post, B., "Building the Business Case for Group Support Technology," Proceedings of the 25th Annual Hawaii International Conference on Systems Sciences, IV, (1992), p. 34-45.

[27] Prescott, M.B., and Conger, S., "Diffusion of Innovation Theory: Borrowings, Extensions, and Modifications from IT Researchers," Unpublished paper, (1995).

[28] Rogers, E.M., *The Diffusion of Innovations*, 3rd Edition, Free Press, New York, 1983.

[29] Ryan, B., and Gross, N.C., "The Diffusion of Hybrid Seed Corn in Two Iowa Communities," *Rural Sociology*, (1943), 8, p. 15-24.

[30] Straub, D., "The Effect of Culture on IT Diffusion: Email and FAX in Japan and the U.S.," *Information Systems Research*, 5, 1, (1994), p. 23-47.

Recycling Decision Trees in Numeric Domains

Miroslav Kubat
Center for Advanced Computer Studies
University of Southwestern Louisiana, Lafayette, LA 70504-4330, U.S.A.
and Department of Computer Science
Southern University in Baton Rouge, Baton Rouge
E-mail: mkubat@cacs.usl.edu

Keywords: decision trees, context, second tier

Edited by: Rudi Murn
Received: May 6, 1999 **Revised:** November 11, 1999 **Accepted:** January 13, 2000

A decision tree's classification performance can drop if the tree is used in a changed context such as different accent in speach recognition. This brittleness can partially be rectified by the use of a cheap second tier implemented as a linear classifier. The transfer of the tree to a novel context is accomplished by re-inducing the second tier, without the need to re-induce the more expensive first tier. Experiments reported in this paper indicate that quick adaptation to the target context can indeed be achieved.

1 Introduction

Even a good classifier can lose much of its performance when transferred to a new context, such as different accent in phoneme recognition or novel font in character identification. Context sensitivity of machine-learning concepts was reported by Katz, Gately, and Collins (1990), Turney (1993a, 1993b), Waltrous (1993), Waltrous and Towell (1995), Widmer (1996), and some others. An altered context can change attribute scales, thresholds, and even the relevance of attributes.

This paper focusses on the following task. A decision tree was induced from examples provided by a *source* context, say, the British accent in natural-language understanding. Later, the tree is to classify examples in a *target* context, say, the American accent. None of the examples of the target was seen during the induction phase. If the two contexts significantly differ, the classifier's accuracy in the target will drop to levels that call for action. But *what* action really?

In principle, one can re-run the learning program on the target examples and dispose of the previous decision tree for good. But this is hardly satisfactory. The fact that the agent does not capitalize on previous experience is at variance with common sense. Laborious re-learning from scratch is expensive: with applications scaling up, some researchers have experimented with domains where even Quinlan's (1993) C4.5 runs for several days (Musick, Catlett, and Russell, 1993). Computational efficiency is an issue in data mining with terabytes of data in store and millions of transactions handled each day.

To avoid the need for re-learning after a *moderate* context change, the engineer is advised to choose an appproppriate bias (Gordon and desJardings, 1995). However, how to determine the bias when the future contexts are unknown? One possibility is premeditated overgeneralization: a pruned decision tree will be less context sensitive because the target might differ from the source in the-out features that have been pruned out.

Pratt, Mostow, and Kamm (1991) encourage investigation of methods for *transfer of knowledge*, an inexpensive way to "recycle" existing knowledge. They implemented a system that used target-context examples to update a neural net whose initial architecture was determined in the source. Experimental studies of Pratt (1993, 1996) indicate that this strategy offers significant savings in the neural net re-training. The idea of transfer also bears on *life-long learning* (Thrun and Mitchell, 1993; Thrun, 1996) where new concepts are induced using the knowledge from previous learning tasks, thus reducing the number of examples needed to achieve certain performance.

The impact of context change has also been addressed in the field of on-line learning where the issue is commonly referred to as *concept drift*. Several methods to cope with this phenomenon exist (Schlimmer and Granger, 1986; Kubat, 1989; Widmer and Kubat, 1996), and the problem has been subjected to in-depth theoretical analyses (Kuh et al., 1991, 1992; Helmbold and Long, 1994). However, lessons from on-line learning do not straightforwardly extend to batch learning because the two paradigms often use different concept representation.

Anticipating the problem more than a decade ago, Michalski (1987) suggested to separate the concept's core from its contextual aspects: to discern the concept *representation* (first tier) from context-dependent *interpretation* (second tier). The approach was further elaborated by Michalski (1989, 1990) and Zhang (1991), reached its high point in the work of Bergadano et al. (1992), and has been adopted also by the neural-network community (Sun, 1995). The system proposed by Baxter (1995) conceived

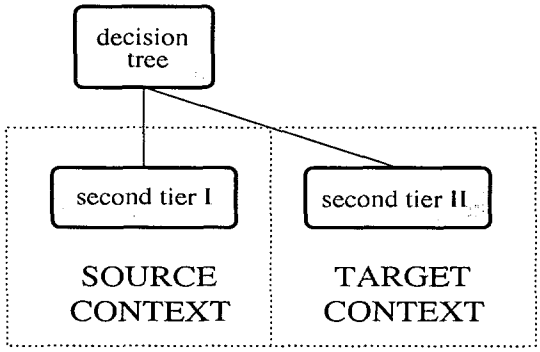


Figure 1: The decision tree with a second tier is transferred from the source context into the target context. The adaptation is carried out by re-induction of the second tier without the need to change the tree.

the second-tier as a small neural net superposed on a much larger first-tier network: whereas the first tier is fixed, each context has its own second tier. Recently, Kubat (1996) implemented a simple second tier as an extension to a decision tree, and, among other things, mentioned that, under circumstances, the second tier might facilitate inexpensive recycling of the decision tree.

This last idea motivated the experiments reported in this paper. The question is stated as follows. Suppose that a decision tree has been provided with a linear second tier as in Kubat (1996). In the event of a changed context, can this classifier be adapted simply by re-inducing the cheap second tier (instead of the more expensive first tier) as illustrated by Figure 1?

2 Task Definition

The paper deals with learning to recognize *numeric* concepts. The agent learns from examples expressed as pairs $[\mathbf{x}, c(\mathbf{x})]$, where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is a vector of attributes, and $c(\mathbf{x})$ is the corresponding concept. A set of concepts can be thought of as a function $c : R^n \rightarrow \{a_1, a_2, \ldots\}$, where $a_1, a_2, \ldots$ are *concept labels*. This function defines on R^n a set of regions. The same concept label can be assigned to one or more regions. The learner's task is to find another function, called a *classifer*, $h : R^n \rightarrow \{a_1, a_2, \ldots\}$, minimizing the probability that $h(\mathbf{x}) \neq c(\mathbf{x})$ for any $\mathbf{x}$ drawn randomly from a fixed distribution.

With the transition from the source to the target the function c changes and the performance of a classifier can drop. The extent of this drop is a measure of the context change between the source and target. By contrast to concept *drift*, that refers to gradual context change, this paper will focus on an abrupt context *switch:* from a certain moment, all objects presented for classification come from the target context.

In this paper, the first tier is implemented as a decision tree induced in the source by C4.5, and will remain unchanged after the switch. The "tailoring" to the target context is carried out by the second tier that has the form of a linear classifier. Induction of linear classifiers is cheaper than induction of decision trees. For n attributes and m examples, Murthy, Kasif, and Salzberg (1994) show that the complexity of the induction of a single node of their OC1 (a linear classifier, in fact) is $O(nm \log n)$, whereas C4.5's complexity is $O(nm^2)$. Likewise, the complexity of the delta rule (Widrow and Hoff, 1960) for a fixed number of epochs is $O(nm)$.

The fact that induction of linear classifiers is cheaper than induction of decision trees promises computational savings in the two-tiered scenario. However, will the classification performance of the two-tiered classifier be satisfactory, too? The rest of the paper endeavors to provide experimental evidence to this claim.

3 Decision Tree with a Linear Second Tier

The classification strategy used in decision trees has remained virtually unquestioned for decades. Each path from the root to a leaf defines *rule* whose antecedent is a conjunction of tests along this path, and the consequent is the label associated with the given leaf. Quinlan (1987) suggested to optimize these rules by the removal of some of their tests. Another strategy, adopted by Helmbold and Shapire (1995), was inspired by the question how to adjust properly the extent of tree pruning. They consider all possible prunings in parallel, and treat them as voting "experts." To increase flexibility, Carter and Catlett (1987) replaced the crisp tests in numeric domains ($x < \theta$) by piecewise linear functions, arguing that crisp tests fail to discern how close the attribute value is to the threshold: the output of the test $x_i > 5$ will be the same whether $x_i = 5.01$ or $x_i = 1000$. Piecewise linear tests provide some tolerance in the vicinity of the thresholds.

The distinctive feature of DT-2T is the fact that, in addition to inducing the decision tree (using Quinlan's C4.5), it also generates the second tier. The program DT-2T employs soft thresholding implemented by the sigmoid function $f(x_i) = \frac{1}{1+e^{-|x_i-\theta|}}$, where θ is the threshold. Note that $f(\theta) = 0.5$, and that $f(x_i)$ converges to its limits (0 or 1) only when x_i is sufficiently distant from the threshold θ.

The tree-to-rules transformation is illustrated by Figure 2. Each path from the root to a leaf is turned into an if-then rule whose antecedent is a conjunction of the softened tests along the path. Each test acquires two distinct forms, $x_i < \theta_j$ (left branch) and $x_i \geq \theta_j$ (right branch). If the output of the j-th test in the k-th rule is denoted as r_{jk}, then the proximity of the k-th rule to the example is calculated as the product $\Pi_{j=1}^{T_k} r_{jk}$, where T_k is the number of tests in the k-th rule. As it is common in the pattern-recognition literature, one "void" rule whose proximity is fixed at 1 is provided.

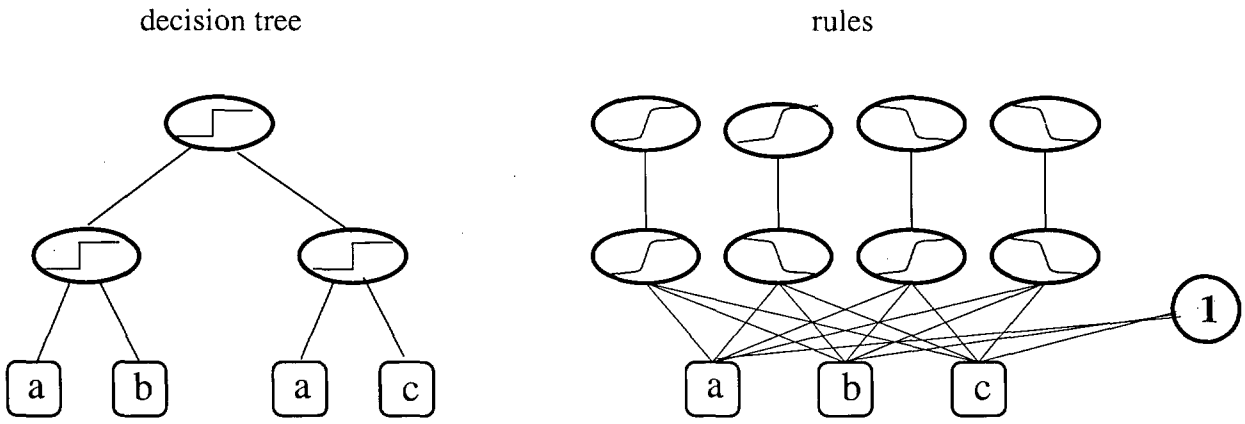


Figure 2: The decision tree is interpreted as a set of rules with "soft" tests. The second tier weighs the impact that each rule has on the individual concept labels.

Examples are thus *transformed* from the original R^n-space to the n_R-dimensional space (for n_R rules) where the i-th attribute gives the example's proximity example to the i-th rule. The next step is to induce a linear classifier for these new data. Denote by $\mathbf{X}$ the matrix whose j-th row represents the j-th (transformed) example, and the i-th column represents the example's proximity to the i-th rule. Denote by $\mathbf{C}$ the matrix such that if the j-th example is labeled with the i-th concept, then the i-th value in the j-th row of $\mathbf{C}$ is 1 and all other values in this row are -1. The matrix of weights, $\mathbf{W}$, should minimize the mean square error $\mathbf{E} = \mathbf{X} \cdot \mathbf{W} - \mathbf{C}$. For implementational convenience, DT-2T uses the technique of a *pseudoinverse matrix*, building on the well-known fact that the mean square error is minimized when $\mathbf{W} = (\mathbf{X^T X})^{-1}\mathbf{X^T C} = \mathbf{X^P C}$.

The rules are fully interconnected with the concept labels, and these interconnections are weighted. Rules that are less relevant for the given concept will carry smaller weights. If a rule suggests that the example represents concept a_1, then the weight between this rule and a_1 will be positive; the weight between this rule and the other concept labels will tend to be smaller, perhaps even negative. Each output unit corresponds to one concept. The system classifies an example with the concept label of the unit giving the highest output.

Figure 3 illustrates the behavior of the linear second tier for the simple case where there are just three rules and two concept labels. Each of the three coordinates represents the proximity of an example to a rule. A decision tree with crisp tests will send all examples to the vertices $(1,0,0), (0,1,0)$ and $(0,0,1)$. With soft thresholds, the examples become more "scattered." The hyperplane in the second tier makes it possible that also some examples lying between the vertices can be correctly classified. Consider two rules that differ only in the last test that precedes the leaves—the test has the form of $x_i < \theta$ in one rule and $x_i \geq \theta$ in the other rule. If the context switch manifests itself by an altered value of θ, then the examples that were originally located at the two corresponding vertices move away from the vertices. The system compensates for this change by tilting the hyperplane accordingly.

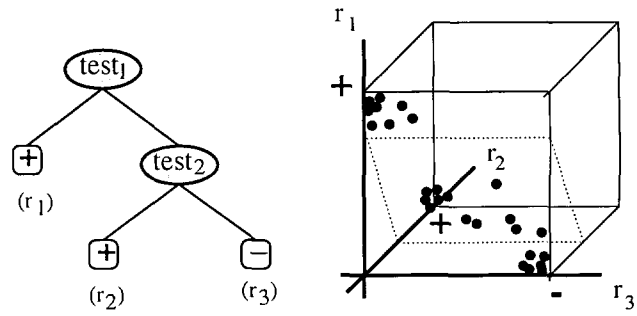


Figure 3: When re-described in terms of proximities to each of the tree branches, the examples tend to cummulate in the vicinity of the vertices. The hyperplane of the second tier helps improve classification of ambiguous examples.

4 Preliminary Experiments with Synthetic Data

For initial insight, toy data files with positive and negative examples were synthesized. Attribute values were randomly generated according to the *normal distribution* where each class is defined by the mean vector, μ, and by the vector of standard deviations, σ. Figure 4 shows the *source* domain. The positive class is defined by $\mu_+ = [0,0]$ and $\sigma_+ = [1,1]$, whereas the negative class is defined by $\mu_- = [2,0]$ and $\sigma_- = [2,2]$. Formal analysis would show that the ideal bayesian classifier for this task is a circle centered at $[-2/3, 0]$, with radius $r = 2.34$. Its performance is 81.51%.

For source, as well as for each target, 1000 examples were generated, 600 of them being used for learning, the rest for testing. The positive and negative examples were equally represented in all the training and testing sets. The decision trees were pruned by C4.5's default. Two methods for creating the target were used. In case A, a "linear" shift,

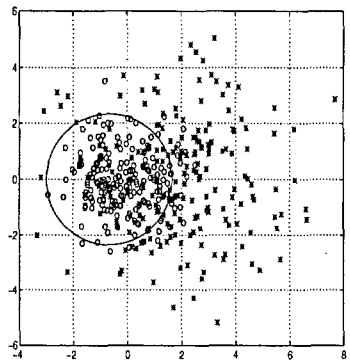


Figure 4: Gaussian data. The ideal decision surface is circular.

the disparity between source and target was modeled by shifted centers. In case B, a "non-linear" shift, the disparity was modeled by altered standard deviations.

Table 1 shows the results for case A. The first two rows are meant as reference points: *C4.5: re-used* shows how much C4.5's performance drops with context change (indicating the degree of discrepancy between source and target) and *C4.5: re-learned* shows the performance of a decision tree re-induced in the target. The remaining rows characterize the behavior of DT-2T. The row *DT-2T: re-used* shows how much the source classifier suffers from a mechanical transfer, whereas *DT-2T: re-learned* provides the results of a classifier that was trained and tested in the same context. The headings define each distribution by its mean (the bracketed expression) and standard deviation. The standard deviation and the distance between the centers of the positive and negative concepts are unaffected by the transfer, which ensures that C4.5, run separately in the source and target, induces decision trees with the same topology, only with changed thresholds. The accuracy achieved by re-learned C4.5 (and re-learned DT-2T) is the same in each target. As the centers move further from the original position, the utility of re-used C4.5 (and re-used DT-2T) drops.

The reader can see that the values in *DT-2T: post-tuned* are virtually unaffected even by the extremely strong disparity between the source and target 4. This indicates that in the testbed A, re-induction of the second tier will adjust the tree to any shift of the concept centers, provided that the distance between the centers remains unchanged. Supplementary experiments showed that reasonable recovery was possible even when the distances between the centers changed.

Table 2 shows that similar behavior can be observed also in case B (altered standard deviations): post-tuned DT-2T achieves the same performance as re-learned DT-2T. In two of the three targets, post-tuned DT-2T outperforms re-learned C4.5. Target 2 is difficult because here the decision surface is linear, with the theoretical performance 66% (the two concepts heavily overlap).

Note that in target 1 the performance of re-used classifiers improved. This is because the negative region in target 1 is fairly compact: many false negatives from the source will find themselves in the positive region of the target. This is an important observation: performance of re-used classifiers depends not only on the extent of context change, but also on the separability of the classes in the target context.

The decision trees generated by C4.5 in the gaussian domain typically contain dozens of tests that transform, non-linearly, the original examples. This explains the performance: non-linear transformation has long been known to increase linear separability of the data (Cover, 1965).

5 Two Case Studies

Synthetic domains permit the researcher to control characteristics he or she believes are relevant for the studied phenomenon. To what extent these characteristics are *realistic* is another question, and experiments with synthetic data thus have to be supplemented with real-world case studies.

5.1 Phoneme Recognition

The repository of machine-learning tasks at the University of California, Irvine (Murphy and Aha, 1995), contains a data file recorded during a phoneme-recognition study (Robinson, 1989). Each example represents an audial spectrum of a single utterance described by 10 numeric attributes and labeled with one out of 11 different vowels. In the phoneticians' terminology, the vowels are referred to as hid, hId, hEd, hAd, hYd, had, hOd, hod, hUd, hud, and hed. The examples were recorded from 15 different speakers: 4 males and 4 females in the training set, and 4 males and 3 females in the testing set. Each of these persons provided 6 instances of each vowel, which amounts to 66 examples per speaker. The total number of training examples is 528 and the total number of testing examples is 462.

The obvious context in this domain, the speaker's gender, suggests the following scenario. Suppose that the source decision tree was induced from female examples. After some time, the tree is used to identify phonemes in males. Alternatively, the target can be represented by a mixed population of males and females. The original file was thus divided into four subsets: male-training, female-training, male-testing, and female-testing. For each source (male of female), two different targets were considered: the opposite gender and mixed population.

Table 3 summarizes the results. Since the decision surface is highly non-linear, and the concepts overlap, the performance of all learners is poor (for comparison, linear regression scored 33.3% on the mixed data). The accuracy of re-learned C4.5 in males (48.9%) is better than in females (30.3%), which indicates greater variance in the female examples. The apparent increase in accuracy in re-used C4.5 (trained in females, re-used in males) is thus actually a reduction. In DT-2T, this phenomenon is less pronounced.

Table 1: Gauss A. The target contexts have shifted means.

	source	target 1	target 2	target 3	target 4
negative examples	([0,0],1)	([1,0],1)	([2,0],1)	([4,0],1)	([10,5]),1)
positive examples	([2,0],2)	([3,0],2)	([4,0],2)	([6,0],2)	([12,5]),2)
C4.5: re-used	78.0	75.0	60.8	48.0	47.5
C4.5: re-learned	78.0	78.0	78.0	78.0	78.0
DT-2T: re-used	81.0	75.3	61.3	47.5	52.5
DT-2T: re-learned	81.0	81.0	81.0	81.0	81.0
DT-2T: post-tuned	81.0	80.8	80.5	80.8	81.0

Table 2: Gauss B. The target contexts have altered standard deviations.

	source	target 1	target 2	target 3
negative examples	([0,0],1)	([0,0],0.5)	([0,0],2)	([0,0],2)
positive examples	([2,0],2)	([2,0],2)	([2,0],2)	([2,0],1)
C4.5: re-used	78.0	84.8	55.5	64.2
C4.5: re-learned	78.0	92.5	65.2	77.0
DT-2T: re-used	81.0	87.0	59.8	62.5
DT-2T: re-learned	81.0	94.8	64.0	80.5
DT-2T: post-tuned	81.0	94.5	65.5	80.5

Table 3: Transfer between genders in vowels (11 classes)

	src: f	trgt: m	trgt: all
C4.5: re-used	30.3	36.4	33.8
C4.5: re-learned	30.3	48.9	42.6
DT-2T: re-used	47.0	36.4	40.9
DT-2T: re-learned	47.0	50.0	48.5
DT-2T: post-tuned	47.0	53.4	52.8
	src: m	trgt: f	trgt: all
C4.5: re-used	48.9	17.2	35.3
DT-2T: re-used	50.0	22.2	38.1
DT-2T: post-tuned	50.0	44.4	46.3

Mechanical transfer of C4.5 or DT-2T from males to females is useless. For instance, the accuracy of the tree induced by C4.5 drops from 48.0% to 17.2%. Importantly, the results of post-tuned DT-2T are only slightly worse than those of re-learned DT-2T in the male-to-female transfer, and are even better in the female-to-male transfer.

Detailed examination of C4.5's output revealed that some vowels were more difficult to discern. To study the learner's behavior on these "difficult" vowels, two subdomains were extracted: VW-012 with examples of hid, hId and hEd; and VW-345 with examples of hAd, hYd, and had. As before, each of the subdomains was divided into training and testing examples, subdivided into males and females.

The results are shown in Tables 4 and 5. Again, mechanical use of the source classifier in the target (C4.5 re-used and DT-2T re-used) gives poor performance. Interestingly, in VW-345 post-tuned DT-2T outperforms re-learned DT-2T in 3 out of 4 transfers (female-to-male, female-to-all, and male-to-female). In VW012, this happens in 2 out of 4

Table 4: Transfer between genders in VW-012 (3 classes)

	src: f	trgt: m	trgt: all
C4.5: re-used	61.1	23.6	39.7
C4.5: re-learned	61.1	68.1	64.3
DT-2T: re-used	61.1	44.4	51.6
DT-2T: re-learned	61.1	72.2	66.7
DT-2T: post-tuned	61.1	54.2	73.0
	src: m	trgt: f	trgt: all
C4.5: re-used	68.1	33.3	53.2
DT-2T: re-used	72.2	33.3	55.6
DT-2T: post-tuned	72.2	55.6	51.6

Table 5: Transfer between genders in VW-345 (3 classes)

	src:f	trgt: m	trgt: all
C4.5: re-used	48.1	50.0	49.2
C4.5: re-learned	48.1	58.3	47.6
DT-2T: re-used	61.1	56.9	58.3
DT-2T: re-learned	61.1	69.4	51.6
DT-2T: post-tuned	61.1	72.2	65.9
	src: m	trgt: f	trgt: all
C4.5: re-used	58.3	48.1	54.0
DT-2T: re-used	69.4	48.2	60.3
DT-2T: post-tuned	69.4	51.9	59.5

transfers (female-to-all and male-to-female). Although this can be mere coincidence, one can stipulate that the knowledge acquired in the source helps the learner in the sense that post-tuning with smaller target training sets now yields higher performance than in the re-learned case. However, the small number of examples in this subdomain makes it unrealistic to investigate the learning curves.

Statistical data analysis provides additional insight. Apart from the altered variance, the context switch causes shifts in some attribute values. For instance, the values of attributes $1, 6, 7$, and 8 in VW-012 are greater in males than in females with confidence level 99% according to t-test, whereas attributes $3, 4, 5$, and 10 have greater values in females (with the same confidence). As another example, attribute 9 in female examples of VW-012 has greater values for hid than for hId with confidence level 95%, but the same attribute in males has greater value for hId than for hid with confidence level 90%. This shift is much less pronounced in VW-345, which seems to corroborate the above hypothesis explaining why post-tuned DT-2T outperformed re-learned DT-2T more often in VW-345 than in VW-012.

5.2 Identification of Sleep Stages

The task in the second case study is to identify distinct stages of human sleep based on 15 numeric attributes such as EEG amplitude, heart rate, or respiration. Manual classification of recordings of a 8-hours sleep is expensive, requiring several hours' effort of a highly qualified professional. Early attempts to automate this process by machine learning (Kubat, Pfurtscheller, and Flotzinger, 1994) revealed that the involved signals are sleeper-dependent: a decision tree induced from one subject cannot be used to classify another subject. Therefore, a weaker objective was specified: the expert classifies a *subset* of examples obtained from a sleeper, the system learns from them, and only then classifies this sleeper automatically. Even this weaker scenario yields significant savings in the expensive human expertise. The experiments discussed below use three files, `br`, `ra` and `kr`, containing 920, 779, and 931 examples, respectively, labeled with 7 distinct concepts. Domain-specific knowledge and the order of example recording are ignored. The three sleepers represent three contexts.

The first experiment will demonstrate that the second tier facilitates successful transfer between sleepers. Each file was randomly split into disjoint training (60%) and testing (40%) sets. The results in Table 6 confirm that induction of a new second tier is a legitimate option in domains where "re-use" fails. Post-tuned DT-2T achieves similar results as re-learned DT-2T, and outstrips re-learned C4.5. The results of post-tuned DT-2T on `ra` (77.9% and 78.9%) even surpass those of re-learned DT-2T on the same sleeper (76.9%).

Figure 5 demonstrates that post-tuned DT-2T is a good match to re-learned DT-2T in each sleeper. When 30–50% of the examples from the domains `kr` and `ra` are used for post-tuning, then the assertion that post-tuned DT-2T and re-learned DT-2T achieve comparable results is true with confidence 95% according to Dietterich's (1996) "5x2cv" test. Although the knowledge acquired in the source does not improve the learning curves of the post-tuned classifier in the target, the graphs show that the source decision tree can successfully be recycled.

Table 6: Transfer between sleepers: initial experience

	br	ra	kr
C4.5: re-learned	79.3	76.3	61.0
DT-2T: re-learned	82.6	76.9	67.7

	src: br	trgt: ra	trgt: kr
C4.5: re-used	79.3	42.0	44.1
DT-2T: re-used	82.6	48.7	24.7
DT-2T: post-tuned	82.6	77.9	66.1
	src: ra	trgt: br	trgt: kr
C4.5: re-used	76.3	14.9	24.7
DT-2T: re-used	76.9	51.9	31.7
DT-2T: post-tuned	76.9	81.5	68.0
	src: kr	trgt: br	trgt: ra
C4.5: re-used	61.0	35.6	19.6
DT-2T: re-used	67.7	32.6	4.9
DT-2T: post-tuned	67.7	82.1	78.9

6 Discussion

The purpose of this paper was to verify the idea that a linear second tier added to a decision tree can facilitate efficient recycling of this decision tree in novel contexts. Linear classifiers are computationally less demanding than induction of decision trees, and it is thus cheaper to provide an existing decision tree with a new second tier than to induce a new tree from scratch. Experimental evidence was obtained from a synthetic gaussian domain and from case studies from phoneme recognition and sleep classification domains. The fact that existing decision trees can be recycled can prove useful in many realistic domains. For instance, data mining applications often require that *many* similar concepts be learned. The experience reported in this paper suggests to induce just a few generic decision trees, each tree representing two or more concepts to be further distinguished by the cheaper second tiers.

The described system can be improved along several dimensions. First, the technique used for the induction of the second tier (pseudoinverse matrix) is known to be sensitive to noise and outliers. Among possible alternatives, the algorithm employed in the system OC1 (Murthy, Kasif, and Salzberg, 1994) is attractive because it minimizes the number of involved attributes. In DT-2T, this would mean to connect each rule only to those concept labels that are really relevant. The understanding of the individual tree paths as voting experts suggests to exploit the results of the

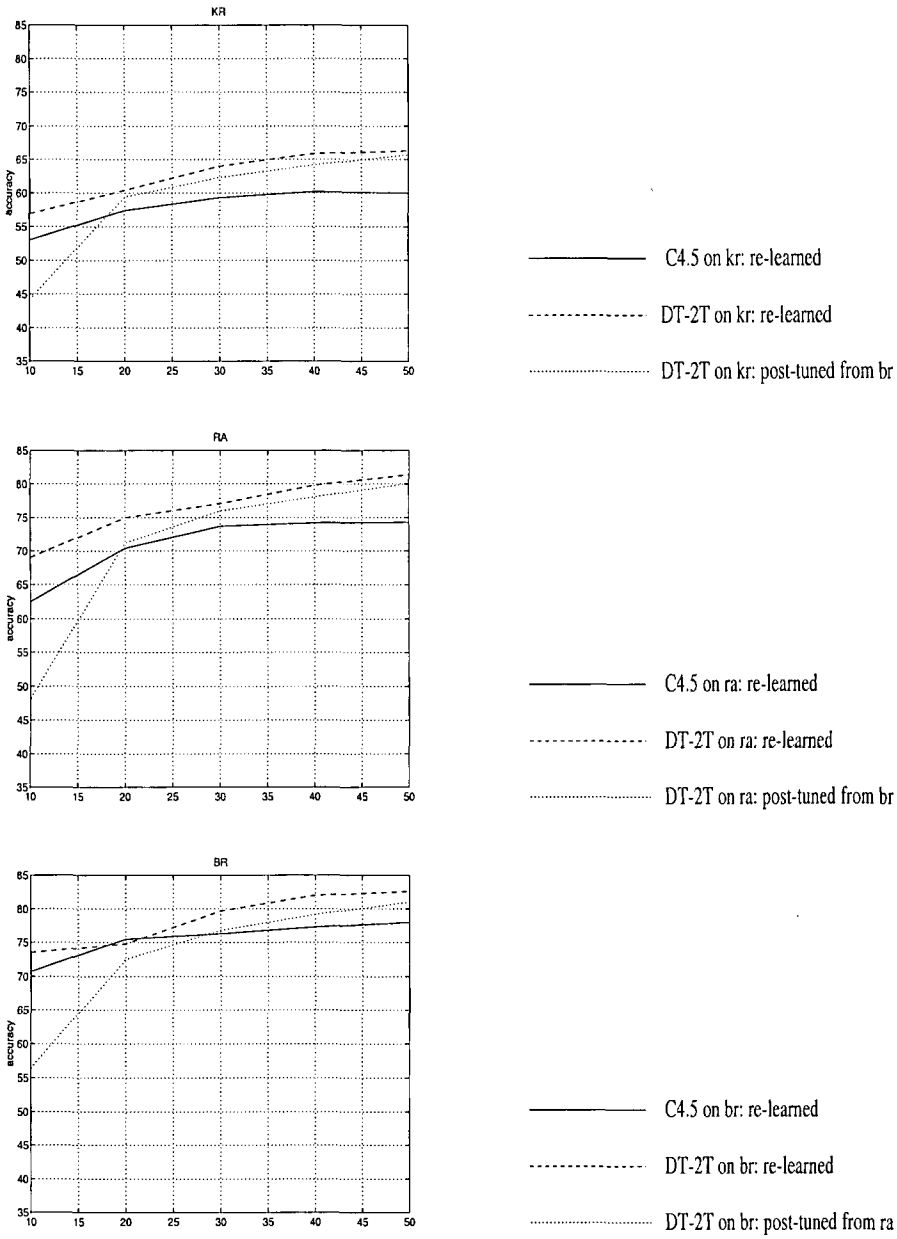


Figure 5: Learning curves characterizing the learning speed of DT-2T.

research in combining expert opinions. The analyses provided by Cesa-Bianchi et al (1993); Littlestone and Warmuth (1994); and Kearns and Seung (1995) prove the computational effectiveness of the requisite algorithms. As an alternative to the current method of "softening" the decision tree, the technique presented by Kubat and Ivanova (1995) can be recommended: the hyperrectangular regions defined in the instance space by the induced decision trees are replaced with gaussian kernels.

The experiments reported in this paper focused exclusively on *univariate* decision trees. Similar techniques might prove successful also in *multivariate* trees (Brodley and Utgoff, 1995; Murthy, Kasif, and Salzberg, 1994; Kubat and Flotzinger, 1995). The second-tier approach can be employed to recycle other expensive classifiers such as multilayer perceptrons (Rumelhart and McClelland, 1986), RBF networks (Broomhead and Lowe, 1988), or perhaps even in inductive logic programming (Lavrač and Džeroski, 1994). From the perspective of recycling, each of these paradigms has its own idiosyncracies that call for further investigations.

The price for DT-2T's performance is decreased understandability. The explanations provided by the original decision tree make sense only when the tree suggests the same concept label as the complete DT-2T. Other examples remain unexplained. However, understandability of numeric decision trees with dozens of tests (as in the gaussian data and in sleep classification) is not persuasive, either, and the loss in the explanation power caused by the second tier can thus be tolerated.

Acknowledgements

Thanks to Stan Matwin, Peter Turney, Rob Holte, and Gerhard Widmer for the countless discussions about contextual aspects in learning. The sleep data belong to the Department of Medical Informatics, Technical University in Graz, Austria, and have been recorded and classified under a grant sponsored by 'Fonds zur Förderung der wissenschaftlichen Forschung' (Project S49/03). Thanks to Gert Pfurtscheller for his kind permission to use them.

References

[1] Baxter, J. (1995). Learning Internal Representations. *Proceedings of the 8th Annual Conference on Computational Learning Theory* (pp. 311–320), Santa Cruz, CA

[2] Bergadano, F., Matwin, S., Michalski, R.S., and Zhang, J. (1992). Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System. *Machine Learning*, 8, 5–43

[3] Brodley, C.E., & Utgoff, P.E. (1995). Multivariate Decision Trees. *Machine Learning*, 19, 45–77

[4] Broomhead, D.S. and Lowe D.(1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2, 321–355

[5] Cesa-Bianchi, N., Freund, Y., Helmbold, D.P., Haussler, D., Schapire, R.E., and Warmuth, M.K. (1993). How to Use Expert Advice. *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing* (pp. 382–391)

[6] Carter, C. and Catlett, J. (1987). Assessing Credit Card Applications Using Machine Learning, *IEEE Expert*, Fall issue, 71–79

[7] Cover, T.M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, EC-14, 326–334

[8] Dietterich, T.G. (1996). Statistical Tests for Comparing Supervised Classification Learning Algorithms. Technical Report, Department of Computer Science, Oregon State University

[9] Gordon, D.F. and desJardins, M. (1995). Special Issue on Bias Learning. *Machine Learning*, Vol. 20

[10] Helmbold, D.P. and Long, P.M. (1991). Tracking Drifting Concepts Using Random Examples. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory (COLT-91)*, Santa Cruz, CA, pp. 13–23.

[11] Helmbold, D.P. and Shapire, R.E. (1995). Predicting Nearly as Well as the Best Pruning of a Decision Tree. *Proceedings of the 8th Annual Conference on Computational Learning Theory* (pp. 61–68), Santa Cruz, CA

[12] Katz, A.J., Gately, M.T., and Collins, D.R. (1990). Robust Classifiers Without Robust Features. *Neural Computation*, 2, 472–479

[13] Kearns, M. and Seung, S. (1995). Learning from a Population of Hypotheses. *Machine Learning*, 18, 255–276

[14] Kubat, M. (1989). Floating Approximation in Time-Varying Knowledge Bases. *Pattern Recognition Letters* 10, 223–227.

[15] Kubat, M. and Flotzinger, D. (1995) Pruning Multivariate Decision Trees by Hyperplane Merging. *Proceedings of the European Conference on Machine Learning*, Heraklion, Crete, Greece, 190–199

[16] Kubat, M. and Ivanova, I. (1995). Initialization of RBF Networks with Decision Trees. *Proceedings of the 5th Belgian-Dutch Conference on Machine Learning, BENELEARN'95*, Brussels, Belgium, 61–70

[17] Kubat, M. (1996). Second Tier for Decision Trees. *Machine Learning: Proceedings of the 13th International Conference*, Morgan Kaufmann Publishers, San Francisco, CA

[18] Kubat, M., Pfurtscheller, G., and Flotzinger D. (1994). AI-Based Approach to Automatic Sleep Classification. *Biological Cybernetics*, 79, 443–448

[19] Kuh, A., Petsche, T. and Rivest, R.L. (1991). Learning Time-varying Concepts. In *Advances in Neural Information Processing Systems (NIPS)* 3, pp.183–189. San Mateo, CA: Morgan Kaufmann.

[20] Kuh, A., Petsche, T. and Rivest, R.L. (1992). Incrementally Learning Time-varying Half-planes. In *Advances in Neural Information Processing Systems (NIPS)* 4, pp.920–927. San Mateo, CA: Morgan Kaufmann.

[21] Lavrač, N. and Džeroski, S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, Hertfordhsire

[22] Littlestone, N. and Warmuth, M.K. (1994). The Weighted Majority Algorithm. *Information and Computation*, 108, 212–261

[23] Michalski, R.S. (1987). How to Learn Imprecise Concepts: A Method Employing a Two-Tiered Knowledge Representation for Learning. *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, pp. 50–58

[24] Michalski, R.S. (1989). Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness. In S. Vosniadu and A. Ortony (eds.), *Similarity and Analogy*. Cambridge University Press

[25] Michalski, R.S. (1990). Learning Flexible Concepts: Fundamental Ideas and Methodology. In Y. Kodratoff and R.S. Michalski (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. III, Morgan Kaufmann

[26] Murphy, P. and Aha, D. (1995). UCI Repository of Machine Learning Databases [machine-readable data repository]. Technical Report, University of California, Irvine

[27] Murthy, S., Kasif, S., and Salzberg, S. (1994). A System for Induction of Oblique Decision Trees. *Journal of Artificial Intelligence Research*, 2, 1–32

[28] Musick, Catlett and Russell (1993). Decision Theoretic Subsampling for Induction on Large Databases. *Proceedings of the 10th International Conference on Machine Learning* (pp.212–219), Amherst, MA

[29] Pratt, L.Y., Mostow, J., and Kamm, C.A. (1991). Direct Transfer of Learned Information among Neural Networks. *Proceedings of the 9th National Conference on Artificial Intelligence, AAAI-91* (pp. 584–589), Anaheim, CA

[30] Pratt, L.Y. (1993). Discriminability-Based Transfer Between Neural Networks. In *Advances in Neural Information Processing Systems* 5, S.J. Hanson, C.L. Giles, and J.D. Cowan (eds.), Morgan Kaufmann, San Mateo

[31] Pratt, L.Y. (1996). Transfer Between Neural Networks to Speed Up Learning. *Journal of Artificial Intelligence Research*

[32] Quinlan, J.R. (1987). Generating Production Rules from Decision Trees. *Proceedings of the 4th International Machine Learning Workshop* (pp. 31–37), San Mateo, CA, Morgan Kaufmann

[33] Quinlan J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo

[34] Robinson, A.J. (1989). Dynamic Error Propagation Networks (PhD. thesis), University of Cambridge, Department of Engineering

[35] Rumelhart, D.E. and McClelland, J.L. (1986). *Parallel Distributed Processing*. MIT Bradford Press

[36] Sun R. (1995). A Two-Level Hybrid Architecture for Structuring Knowledge for Commonsense Reasoning. In Ron Sun and Bookman, L.A. (eds.): *Computational Architectures Integrating Neural and Symbolic Approaches: A Perspective on the State of the Art*. Kluwer Academic Publishers, Boston, pp. 247–281

[37] Thrun, S.B. (1996). *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic Publishers, Boston

[38] Thrun, S.B. and Mitchell, T.M. (1993). Lifelong Robot Learning. In Steels, L. (Ed.), *Proceedings of the NATO ASI: The Biology and Technology of Intelligent Autonomous Agents.*

[39] Turney, P.D. (1993a). Exploiting Context when Learning to Classify. *Proceedings of the European Conference on Machine Learning* (pp. 402–407), Vienna, Austria

[40] Turney, P.D. (1993b). Robust Classification with Context-Sensitive Features. *Proceedings of the Sixth International Conference of Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (pp. 268–276), Edinburgh, Scotland.

[41] Waltrous, R.L. (1993). Speaker Normalization and Adaptation Using Second-Order Connectionist Networks. *IEEE Transactions on Neural Networks*, 4, 21–30

[42] Waltrous, R.L. and Towell, G. (1995). A Patient-Adaptive Neural Network ECG Patient Monitoring Algorithm. *Proceedings of Computers in Cardiology*, Vienna, Austria

[43] Widmer, G. (1996). Recognition and Exploitation of Contextual Clues via Incremental Meta-Learning. *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy

[44] Widmer, G. and Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23, 69–101

[45] Widrow, B. and Hoff, M.E. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*, 96–104.

[46] Zhang, J (1991): Integrating Symbolic and Subsymbolic Approaches in Learning Flexible Concepts. *Proceedings of the 1st International Workshop on Multistrategy Learning*, Harpers Ferry, U.S.A., November 7–9, 289–304

Bitmap R-trees

C.H. Ang, S.T. Tan and T.C. Tan
Department of Computer Science, School of Computing
National University of Singapore, Republic of Singapore, 117543
Phone: +65 874 2729, Fax: +65 779 4580
E-mail: {angch, tanst, tantc}@comp.nus.edu.sg

Keywords: spatial data structures, R-tree

Edited by: Rudi Murn
Received: December 3, 1990 **Revised:** January 18, 2000 **Accepted:** February 4, 2000

Bitmap R-tree is a variant of R-tree in which bitmaps are used for the description of the internal and the external regions of the objects in addition to the use of minimum bounding rectangles. The proposed scheme increases the chance of trivial acceptance and rejection of data objects, and reduces unnecessary disk accesses in query processing. It has been shown that with the bitmaps as filters, the reference to the object data file can be cut down by as much as 60%.

1 Introduction

With the widespread use of computers, many non-conventional applications have been developed to handle spatial data in two and three dimensions. The spatial data include map objects such as bridges in a Geographical Information System (GIS), or electronic components such as transistors in a VLSI Computer Aided Design (CAD) application.

With the ever increasing demand for the manipulation of spatial data, spatial database system evolves and many data structures have been proposed such as quadtree [5], kd-tree [4], and R-tree [7]. (Refer to [10] for more details on the various spatial data structures that have been in use.) As one of the earliest proposed tree structures for nonzero-sized spatial objects, R-tree has always been used as a yardstick in assessing the performance of other related data structures. Variants of the R-tree, such as R^+-tree [11], R^*-tree [3], and Rr-tree [9] have been proposed all with the aim to improve performance.

All these structures can be used to support various forms of query, in particular, point query and region query. The bitmap R-tree we propose in this paper aims to enhance the query processing performance of R-tree by extending it with two types of bitmaps. It is hoped that with the use of these bitmaps, some unnecessary disk accesses can be avoided. Even though the construction of the bitmaps requires some processing time, the overall saving in query processing can still be gained.

The paper is organized as follows. We begin with a discussion on R-tree and its variants in section 2, followed by introducing bitmap R-tree in section 3. Query processing using bitmap R-tree is described in section 4 and its performance analysis based on empirical results is presented in section 5. In section 6 we draw our conclusion.

2 R-tree and its variants

R-tree is a multi-dimensional generalization of B-tree [2]. It is used as an indexing structure to speed up the retrieval of spatial objects. It is height-balanced and the insertion and deletion of an object may trigger node splitting and merging.

Usually, a k-dimensional spatial object is fully described with all its spatial and aspatial attributes of interest contained in a long record in a data file. In order to access this record quickly, its offset from the beginning of the file is used as an index which is stored in a leaf node of an R-tree. An entry in a leaf node of an R-tree is a tuple (mbr, oid), where mbr is the k-dimensional minimum bounding rectangle of the object, and oid is the object identifier that can be used to retrieve the full object description record from the file.

Each entry in a non-leaf node of an R-tree is a tuple $(mbr, childptr)$, where $childptr$ is a pointer to a lower level node in the R-tree, and mbr is the minimum bounding rectangle (MBR) that covers all the rectangles in the lower level node pointed to by $childptr$. Figure 1 shows the MBR of an object. Figures 2 and 3 show the planar view of an R-tree, and its corresponding structure, respectively.

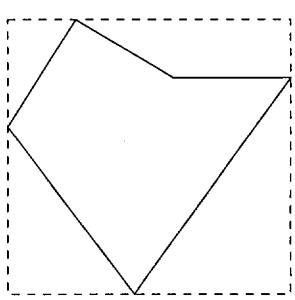


Figure 1: MBR of an object.

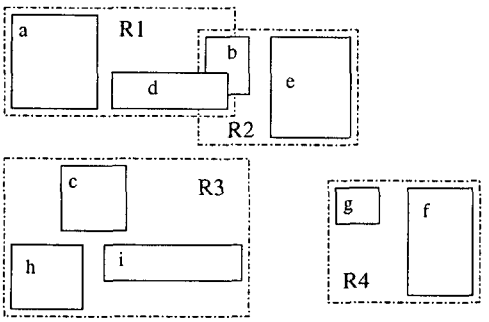


Figure 2: Planar view of an R-tree.

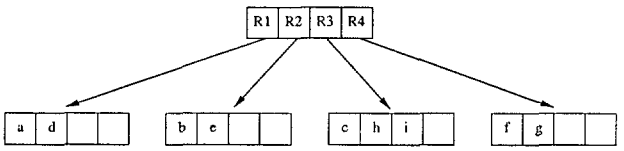


Figure 3: Structure of an R-tree.

Each R-tree node has a predetermined capacity which is the maximum number of entries a node can carry. During the insertion of an entry, a node will be split when it overflows. A new node is allocated, and all entries in the overflowed node together with the new entry are redistributed into the overflowed node and the new node according to some rules. A new index which contains the MBR of the new node and the pointer to the new node has to be inserted into the parent node (the node that contains the pointer to the overflowed node). The overflowed node's MBR has to be updated. The insertion of the new index may in turn cause the parent node to split and the node splitting may propagate upwards. If the propagation reaches the root node, the root node is split, a new root node is created and the depth of the R-tree is increased.

Overlapping regions exist not only in the leaf node but also in the non-leaf nodes of an R-tree. When a query point falls within an overlapping region, there will be multiple access paths leading to the desired objects, resulting in higher search cost [8]. To overcome this problem, R^+-tree was proposed. It uses the object clipping approach to divide objects into as many sub-objects as required so that the MBRs of these internal nodes do not overlap each other. The disjoint partitioning of subspaces ensures a single search path for a given query point.

In Rr-tree, in addition to the use of MBR, the maximum internal rectangle (MIR) is used and stored in the leaf node together with the MBR of an object. The MIR is the largest rectangle that is strictly contained in the object. Any point falling inside the MIR of an object is definitely within the object. The use of MIR can therefore further reduce the number of accesses to the data file, especially in answering a point query. Figure 4 shows the MBR and MIR of an object.

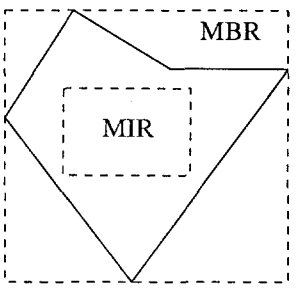


Figure 4: MBR and MIR of an object.

3 Bitmap R-tree

The basic idea of using bitmaps in bitmap R-tree is similar to the Rr-tree except that instead of storing the MIR, the internal and external bitmaps of the object are stored together with its MBR at the leaf node.

A spatial object is usually represented by a simple polygon (a polygon without holes). Two regions are defined by the polygonal boundary of the object: one that is strictly contained in the object and another that is strictly outside the object but still within the MBR. Instead of using 16 bytes for the description of MIR as is done in the Rr-tree, we could make good use of these spaces to describe the two regions in the highest resolution attainable, resulting in the use of two bitmaps, the *internal bitmap* and the *external bitmap*.

Each MBR is divided into 8 by 8 grid cells. A small bitmap can be constructed and stored in 8 bytes if a cell is represented by a bit. The 1s in an internal bitmap of a spatial object represent the corresponding grid cells that are completely within the object while the 1s in an external bitmap represent the corresponding grid cells that are completely outside the object but inside the region confined by the MBR. Figures 5, 6, and 7 show a triangle that is superimposed on an 8x8 grid, its internal bitmap, and its external bitmap. From the internal bitmap and the external bitmap, it is not difficult to obtain the outline bitmap which shows all grid cells that intersect with the boundary of the object. The three bitmaps are closely related; given any two, we can easily derive the remaining bitmap.

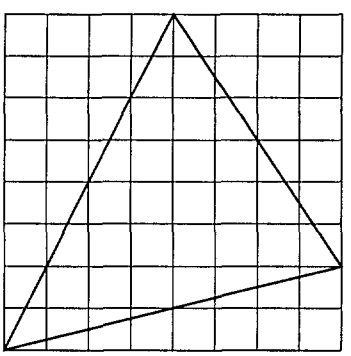


Figure 5: A triangular object T.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 6: Internal bitmap of T.

1	1	1	0	0	1	1	1
1	1	0	0	0	0	1	1
1	1	0	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1

Figure 7: External bitmap of T.

The algorithm to generate the external bitmap of a polygon is adapted from the polygon filling algorithm [6] used in graphics rendering. The outline bitmap is generated by using the voxel traversal algorithm [1] used in ray tracing. With these two bitmaps, the internal bitmap can be obtained.

The deletion and insertion of an object into a bitmap R-tree is the same as that for an R-tree except that the corresponding bitmaps are to be created during the insertion of an entry at the leaf node level.

4 Query processing

In an R-tree, if the query is an arbitrary region, the MBR of the query region will be used to select object entries that satisfy the query MBR. In addition, the object descriptions of all the selected entries are needed for further testing with the query region.

With a bitmap R-tree, the internal and the external bitmaps of the query region, denoted $Qintmap$ and $Qextmap$, can be generated and used in query processing. Using the bitmaps and the MBR of the query region, certain entries may be trivially accepted or rejected without further testing.

Table 1 lists the tests on the bitmaps that are used in processing a query.

In point query, we are to locate all objects that contain a given point. In an R-tree, even though the given point is contained in an MBR, we cannot determine if the object really contains the point. In a bitmap R-tree, if the point is found to be in the $intmap$, then it is inside the object and the object is trivially accepted. If the point is in the $extmap$, then it is outside the object and the object is trivially rejected. Of course, if it is not within the MBR, then it is also outside the object. Only when the given point is found to be in the MBR of an object and is neither in its $intmap$ nor its $extmap$, then will the object description record stored in the data file be accessed.

Test	Result
PointExt	If a query point is mapped into a bit in $extmap$ with value 1, the point is in the external region and the object is trivially rejected.
PointInt	If a query point is mapped into a bit in $intmap$ with value 1, the point is in the internal region and the object is trivially accepted.
BmpNonIntersection	If none of the 0s in $extmap$ is mapped to any of the 0s in $Qextmap$, the regions do not intersect. Otherwise the regions may intersect.
BmpContainment	If all the 0s in the $extmap$ are mapped to 1s in $Qintmap$, it implies that the spatial object of the entry is strictly contained in the query region.
BmpIntersection	If any of the 1s in $intmap$ is mapped onto a 1 in $Qintmap$, the two regions intersect. The object can be trivially accepted for the intersection query.
LineIntersection	Clip the query line to the MBR of the selected entry. Create bitmap of the clipped line and perform BmpIntersection test.

Table 1: Bitmap test functions

In a region query, we want to find all objects covered completely by the given region (a subset query), containing it (a superset query), or simply intersecting it (an intersection query). We only consider intersection query in our study.

In an R-tree, the region specified is usually a rectangle. Any MBR in a leaf node found intersecting with the specified region will cause the corresponding object record to be read in. In a bitmap R-tree, if any portion of the $intmap$ of an object is found to be within the region, the object is trivially accepted. If the region is found to intersect only with the $extmap$ of an object, then the object is trivially rejected. Otherwise, the object record has to be retrieved for further checking.

The use of bitmaps of the given query region simplifies the filtering step. The cost of checking the intersection between the bitmaps of the given region and the bitmaps of the objects is low. Therefore, specifying a non-rectangular query region in a bitmap R-tree will not pose any problem in the performance of query processing.

5 Performance analysis

We implemented R-tree and bitmap R-tree. We did not implement Rr-tree as the area of an MIR is usually smaller than that of the corresponding $intmap$, and with the use

File size	Point query	Line int.	Quadrangle int.
10000	68	69	69
20000	69	65	63
30000	69	64	62

Table 2: Efficiency (in percentage) of bitmaps

Node size	Point query	Line int.	Quadrangle int.
1K	36	27	21
2K	30	20	13
3K	24	15	9
4K	20	5	7

Table 3: Percentages of saving in disk accesses using bitmap R-tree

of the *extmap*, the number of objects that can be trivially accepted or rejected based on the use of both the *intmap* and the *extmap* should be more than that with the use of MIR.

The space requirements of R-tree and bitmap R-tree can be worked out easily. The sizes of an MBR, a child pointer, and a coordinate are 16, 4, and 4 bytes respectively. Thus the sizes of an entry in a leaf node in an R-tree and bitmap R-tree are 20 and 36 bytes respectively. The sizes of the entry in an internal node of both structures are 20 bytes. Assuming that a node is of 1K bytes, then the capacities of a leaf node in an R-tree and a bitmap R-tree are 50 and 28 respectively. The capacity of a non-leaf node in both trees is 50.

The size of the domain from which the spatial objects are drawn is 100000 by 100000. The spatial objects are no larger than 1000 by 1000. The objects are made up of randomly generated points, lines, triangles and quadrangles. Equal number of each type of objects were generated.

During the experiment, the total number of objects trivially accepted and rejected, and the total number of disk accesses were noted. The number of input objects varied from 10000 with increment of 10000 till 30000. The node size varied from 1K bytes to 4K bytes, with increment of 1K bytes at each step. Three different sets of 100 query objects of the same type (point, line, etc.) were randomly generated. The 100 points were used in point query. The line objects were used to locate objects that intersect with the query lines. The quadrangles were also used for intersection query. This would allow us to find out the effect of the shapes of the query objects on the efficiency of the bitmaps in the query processing.

The *efficiency* of the bitmap is defined as t/c, the ratio between t, the number of candidates that are trivially rejected or accepted, and c, the number of candidates selected by checking their MBRs with the query point or with the MBR of the query region. For the same set of test data and the same query set, the efficiency of the bitmaps remained very much stable regardless of the size of the nodes. When the same query set was used, the efficiency varied slightly when different sets of test data were used. This can be seen from Table 2. In point query, 68% of those candidates were either trivially rejected or accepted. For intersection query, the efficiency is 62%.

Although the efficiency of the bitmaps seemed to be quite high, the reduction in disk accesses was about 36% at best. The percentages of saving in disk accesses are shown in Table 3.

The saving is large for point query using small nodes but the advantage of using bitmap R-tree wanes when the node size is increased. This is because a bigger node allows more objects to be stored, hence the MBRs of the internal nodes are larger, more regions overlap, causing more nodes in the lower level to be read in the search. As the capacity of the leaf nodes of the bitmap R-tree is smaller, the same set of candidates is stored in more leaf nodes. Therefore, more leaf nodes have to be accessed even though the efficiency of the filter remains the same.

We have experimented to find out the efficiency of the *intmap* and the *extmap* of the internal nodes. The results showed that they were not effective at all (less than 0.2%) in trimming the access path. This is indeed not surprising. In fact, the *intmap* of an internal node is useless as the low level nodes are still required to be read in when the query point is found within the *intmap*. The saving can only come from the use of the *extmap* of an internal node to terminate the search from proceeding further. When the MBR of an internal node is represented by an 8 by 8 bitmap, each of the 64 grid cells is very likely covered by some of the rectangles of the corresponding entries stored in the low level nodes. This means that the internal bitmap *intmap* is likely to contain all 1s and the external bitmap *extmap* will contain mostly 0s, making the use of bitmaps in an internal node very ineffective.

6 Conclusion

The use of MBR in R-tree helps to filter away objects that are unlikely to be included in the answer to a query. Although the use of MBR is simple, it may be too crude at times, especially when its external region is large.

By using MIR in Rr-tree, even though it helps to filter additional candidates that are definitely to be included in the answer set, it is still not effective when the external region of an object is large and thus the corresponding MIR is small. Since each MIR requires 16 bytes, the capacity of a leaf node in an Rr-tree is significantly reduced.

We propose to enrich the R-tree structure with additional information that can quickly differentiate if a given point is inside or outside an object. With the same storage overhead as MIR by using 16 bytes for storing the internal bitmap and the external bitmap of a given object in bitmap R-tree, we are able to trivially accept or reject more than 60% of the candidates identified based on the MBRs alone. An object with a large external region has a large external bitmap,

and hence this increases its chance of being rejected in a point query or a region query.

Although a bitmap R-tree needs more space and time to process the bitmaps and the tree resulted may be deeper due to the larger number of leaf nodes each of which having fewer entries, the empirical results show that the query performance has generally improved by more than 20% in terms of the number of disk accesses. Since the CPU processing time is only a small fraction of disk access time, a net gain in query performance is evident.

References

[1] John Amanatides and Woo, A fast voxel traversal algorithm for ray tracing, EUROGRAPHICS (1987).

[2] R. Bayer, E. McCreight, Organization and maintenance of large ordered indices, Acta Informatica 1, 3(1972), 173-189.

[3] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, The R^*-tree: An efficient and robust access method for points and rectangles, Proc. of the ACM SIGMOD Conference, Atlantic City (1990), 322-331.

[4] J. L. Bentley, Multidimensional binary search trees used for associative searching, CACM 18, 9(1975), 509-517.

[5] R. A. Finkel, J. L. Bentley, Quad Trees, a data structure for retrieval on composite keys, Acta Informatica, 4(1974), 1-9.

[6] J.D. Foley, A. Dam, S.K. Feiner, and H. F. Hughes, Computer Graphics: Principles and Practice, 2nd edition.

[7] A. Guttman, R-tree: A dynamic index structure for spatial searching, Proc. of the ACM SIGMOD Conference, Boston, (1984), 47-57.

[8] E. G. Hoel, H. Samet, A qualitative comparison study of data structures for large line segment databases, Proc. Int. Conf. on Management of Data (1992).

[9] J. Kim and H. Bae, The design of efficient access method for objects, GIS: Technology and applications, Proc. for the Far East Workshop on Geographic Information Systems, Singapore, 21-22 June 1993, 91-105.

[10] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, (1989).

[11] T. Sellis, N. Roussopoulos, C. Faloutsos, The R^+-tree: A dynamic index for multi-dimensional objects, Proc. 13th International Conference on Very Large Data Bases, Brighton, England, (1987), 507-518.

The Polling Primitive for Computer Networks

Andrzej Czygrinow
Department of Mathematics, ASU, Tempe, AZ, 85287-1804
andrzej@math.la.asu.edu
AND
Michal Karoński
Faculty of Math and CS, AMU, Poznan, Poland
and
Department of Mathematics and Computer Science
Emory University, Atlanta, GA, 30033, USA
karonski@amu.edu.pl, michal@mathcs.emory.edu
AND
Vaidy Sunderam
Department of Mathematics and Computer Science
Emory University, Atlanta, GA, 30033, USA
vss@mathcs.emory.edu

Keywords: distributed computing, polling, hypercube.

Edited by: Marcin Paprzycki
Received: March 12, 1999 **Revised:** November 11, 1999 **Accepted:** February 1, 2000

We describe a distributed computing primitive termed polling that is both a means of synchronization and communication in distributed or concurrent systems. The polling operation involves the collection of messages from nodes in an interconnection network, in response to a query. We define the semantics of polling, and present algorithms for implementing the operation on complete and hypercube networks. Time and message lower bounds are presented, and are followed by an analysis of the number of operations performed at each node for every algorithm. We show that polling in a complete graph on 2^n vertices can be completed in $2n$ rounds using $2^n + 2^{n-3} + \lceil \frac{2^{n-3}+1}{3} \rceil - 1$ messages. In case of n-cube, we show that polling in $2n$ rounds requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{6}\sqrt{2^n} - \frac{4}{3} \rceil$ messages and we present an algorithm that completes polling in $2n$ rounds and sends $2^n + 3 \cdot 2^{n-4} - 1$ messages.

1 Introduction

We define the *polling* operation on interconnection networks as follows: One processor in a network (termed the root) has a "question" that must be asked of all other processors, each of which must respond with an "answer". We wish to perform this operation in minimal time using a minimal number of messages, under the following assumptions:
(1) Processors communicate solely by message-passing; messages may contain the question, one or more answers, or both.
(2) Polling proceeds in "rounds"; a processor may either send or receive at most one message during any round. Processors other then the root may participate only after they have received at least one message (i.e. a message containing the question).

As with most distributed algorithms, the efficiency of polling is measured by the time required to accomplish the operation (number of rounds), and by the number of messages needed. Algorithm goals are to minimize the number of rounds necessary, and given the minimal number of rounds, to minimize the number of messages.

The notion of polling is traditionally associated with terminals and controllers, and with data link protocols. However, polling has important applications in distributed systems, and on non-shared memory multiprocessor machines. Status or resource monitoring, fully replicated queries or updates, the computation of multiple-input functions, and certain synchronization primitives may all be implemented using polling. Polling inherently requires that one processor initiates the operation, that every processor participates, and that all outputs be returned to the initiator. When these are necessary conditions within an application, polling is an effective distributed computing primitive. It seems that until recently, the polling problem received much less attention than the classical broadcasting and gossiping problems for which many results were obtained and different models were studied (see for example [6] and references in it). The time complexity of polling was recently studied by A. Rescigno in [8] and [9]. The communication model considered is however different than

ours. For example, in model in [8] and [9] a node can send a message to all of its neighbors in a single round but it is not possible to send many responses along a single edge. In our model, a vertex can communicate only with one of its neighbors in a single round. On the other hand, we assume that responses can be combined and send as one. Consequently, under the assumptions in [9] polling in a complete graph on n vertices can be done in 2 rounds, in our model it requires about $2\log n$ rounds. The algorithms of [9] are based on special kind of polling trees which are used to distribute the question and to gather the responses. In contrast, our communication graphs are not trees as it is easy to see that if we use any spanning tree as communication graph then the number of messages sent will be twice the number of edges which is greater than what is obtained using our graphs. The number of messages sent is not discussed in Rescigno's papers.

The paper presents the lower bounds and algorithms for polling in networks with complete graph and hypercube topology. In the next section we show the lower bounds and we present an algorithm that performs polling in complete networks in the minimal number of rounds and using the minimal number of messages. Section 3 contains an analysis of polling in the hypercube network. We show a slightly better lower bound for the number of messages and propose a nearly optimal algorithm.

2 Preliminaries

We define a *path* in a network as a sequence of vertices $v_1v_2\ldots v_n$ such that for all $1 \leq i \leq n-1$ there is an edge v_i, v_{i+1}. We then say that the path *covers* n vertices or if v_1 and v_n are already nodes of a different path we say that the path covers $n-2$ new vertices. *Cycle* is defined as a sequence $v_1v_2\ldots v_n$ such that for $1 \leq i \leq n$ there is an edge $v_i, v_{(i+1) \pmod n}$. The *degree of a vertex* is the number of vertices incident to it. Also N will denote the number of nodes in a network, $\lg a$ will denote the logarithm of base 2 and $\ln a$ the logarithm of base e.

We define *partial broadcast* as the delivery of a message originating at the root to a subset of nodes of a network. *Partial gather* is defined analogously as the collection of messages from a subset of nodes of a network.

Proposition 1 *For $0 \leq M \leq 2^n - 1$ partial broadcast to M (gather from M) nodes requires $\lceil \lg (M+1) \rceil$ rounds.*

Proof. We prove the bound in case of partial broadcast. Partial gather is proved in the same way. During one round a node may either send one message to one of its neighbors. Thus number of nodes that have received the message can at most double in each round. Therefore if the number of rounds k is less than $\lg(M+1)$ then the number of nodes which received the message is at most $\sum_{i=1}^{k} 2^i < \sum_{i=1}^{\lg(M+1)} 2^i = M$. Thus $k \geq \lg(M+1)$ and since k must be an integer the proof is complete.

3 The Algorithm for complete graphs

In this section we present the lower bound for the number of rounds and the number of messages. The technique used in the proof of second bound leads to an optimal polling algorithm in complete graphs which is described at the end of the section.

Proposition 2 *Let $N = 2^n - k$ where $0 \leq k \leq 2^{n-1}$.*

(i) If $k = 0$ then the number of rounds is at least $2n$.

(ii) If $0 < k \leq 2^{n-2}$ then the number of rounds is at least $2n-1$.

(iii) If $k \geq 2^{n-2} + 1$ then the number of rounds in is at least $2n-2$.

Proof. (i) Let $N = 2^n$. From Proposition 1 we know that immediately after $n-1$ rounds at most $2^{n-1}-1$ nodes other than root received the message originated at the root. Denote the set of these nodes by L and let $R = V(K_N) \setminus L$. To gather messages from R we need at least $\lceil \lg |R| \rceil = \lceil \lg (2^{n-1}+1) \rceil = n$ rounds. At least one more round is necessary to initiate the participation of nodes in R. The situation is illustrated in Figure 1 (i). This shows that to complete polling in K_N, at least $n-1+1+n = 2n$ rounds are necessary.
(ii) When $k \leq 2^{n-2}$ then $N \geq 2^n - 2^{n-2} = 2^{n-1} + 2^{n-2}$ and Proposition 1 implies that immediately after $n-1$ rounds at most $2^{n-1}-1$ nodes other than root will receive the message originated at the root. To gather from remaining 2^{n-2} nodes, we need $\lceil \lg (2^{n-2}+1) \rceil$ rounds and at least one round to initiate the participation of remaining nodes. Therefore, at least $n-1+1+n-1 = 2n-1$ rounds are required to complete polling in K_N (see Figure 1 (ii)).
(iii) It follows from (i) as $N \geq 2^n - 2^{n-1} = 2^{n-1}$.

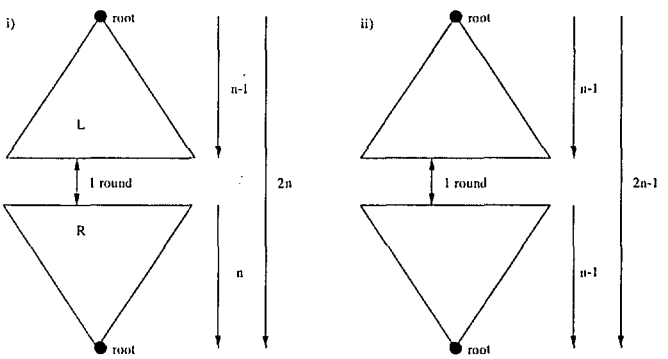


Figure 1: Proof of Proposition 2

Proposition 3 *The polling in a complete graph K_{2^n} in $2n$ rounds requires $2^n + 2^{n-3} + \lceil \frac{2^{n-3}+1}{3} \rceil - 1 \geq 2^n + \frac{1}{3}2^{n-1} - 1$ messages.*

Proof. Consider an optimal polling algorithm with root x. It will be convenient to think about x as a pair of vertices (x_1, x_2) where x_1 sends messages and x_2 receives messages. Let D denote a directed communication graph constructed by the polling algorithm, that is (y, z) is an arc in D if a message is sent from y to z. Let L_i denote the set of nodes that receive a message in the ith round (for example, $L_0 = \{x_1\}, L_{2n} = \{x_2\}$) and set $r = \lceil \frac{2^{n-3}+1}{3} \rceil$. To establish the lower bound for the number of messages we prove the following lemma.

Lemma 4 *For some* $n-1 \leq i \leq n+1$, $|L_i| \geq 2^{n-3} + \frac{2^{n-3}+1}{3}$.

Proof. First observe that due to the requirements of polling $|L_i| \leq 2|L_{i-1}|$ for $i \geq 2$ with the initial conditions $|L_0| = |L_1| = 1$ which gives

$$|L_i| \leq 2^{i-1}. \tag{1}$$

Since the polling requirements for sending and receiving are symmetrical we can reverse the orientation of the arcs of D to obtain a polling algorithm which sends the message from x_2 to x_1. Consequently

$$|L_{2n-i}| \leq 2^{i-1}. \tag{2}$$

Summing over all $1 \leq i \leq n-2$ yields

$$|\bigcup_{i=1}^{n-2} L_i| \leq \sum_{i=1}^{n-2} 2^{i-1} = 2^{n-2} - 1 \tag{3}$$

and

$$|\bigcup_{i=1}^{n-2} L_{2n-i}| \leq 2^{n-2} - 1. \tag{4}$$

Assume now that for all $n-1 \leq i \leq n+1$, $|L_i| < 2^{n-3} + \frac{2^{n-3}+1}{3}$. Then by (3) and (4) the number of vertices in D (counting x_1 and x_2) is less than

$$2 + 2(2^{n-2} - 1) + 3(2^{n-3} + \frac{2^{n-3}+1}{3}) = 2^n + 1,$$

which is a contradiction as D contains $2^n + 1$ vertices. Therefore, for some $n-1 \leq i \leq n+1$,

$$|L_i| \geq 2^{n-3} + \frac{2^{n-3}+1}{3}.$$

Since $|L_i|$ is an integer, Lemma 4 gives that $|L_i| \geq 2^{n-3} + r$ for some $i \in \{n-1, n, n+1\}$. The number of messages sent by the algorithm is equal to the number of arcs in D and so it is enough to prove the following lemma.

Lemma 5 *The number of arcs of* D, $e(D)$ *is at least* $2^n + 2^{n-3} + r - 1$.

Proof. Let i be such that $|L_i| \geq 2^{n-3} + r$. Let D_1 denote the subdigraph of D induced by levels $L_0, \ldots, L_i$ and let D_2 be the subdigraph induced by levels $L_i, \ldots, L_{2n}$. The number of arcs in D is

$$e(D) = e(D_1) + e(D_2),$$

where $e(D_j)$ denote the number of arcs in D_j $(j = 1, 2)$. Since D_1 and D_2 are connected

$$e(D_1) + e(D_2) \geq (2^n + 1) - 2 + |L_i| \geq 2^n + 2^{n-3} + r - 1.$$

Thus, the number of messages sent is at least $2^n + 2^{n-3} + \lceil \frac{2^{n-3}+1}{3} \rceil - 1$.

Next we present the algorithm that completes polling in K_{2^n} in $2n$ rounds and that uses $2^n + 2^{n-3} + r - 1$ messages. The idea is as follows. In the first $n-3$ rounds the greedy procedure is invoked that results in total of $2^n - 2^{n-3} - 1$ nodes covered. The remaining vertices are covered using $\lceil \frac{2^{n-3}+1}{3} \rceil$ paths. More formally, let us define the broadcasting tree of height n, $B(n)$ as follows: $B(0)$ contains just one vertex- the root, for $n > 0$, $B(n)$ is obtained from $B(n-1)$ by adding for each vertex $v \in V(B(n-1))$ exactly one vertex v' and an edge vv'. Note that the vertices of the tree can be grouped into levels, where the ith level contains the vertices that are at distance i from the root. Then the communication graph can be constructed by the following procedure.

Algorithm

1. Take two copies of the broadcast tree $B(n-2)$ of height $n-2$, B and B'. Let $v_1, v_2, \ldots, v_l, (l = 2^{n-3})$ denote the leaves of B and $v_1', v_2', \ldots, v_l'$ the corresponding leaves of B' (both v_i and v_i' correspond to the same vertex of $B(n-2)$). For every $1 \leq i \leq l$ connect v_i with v_i' by a path of length four. The resulting graph looks as in Figure 2.

2. Let $r = \lceil \frac{2^{n-3}+1}{3} \rceil$. To simplify the exposition we assume that $\frac{2^{n-3}+1}{3}$ is an integer. Take r vertices $w_1, \ldots, w_r$ in B that are not leaves and connect w_i with the corresponding vertex w_i' in B' by a path $w_i x_{1i} x_{2i} x_{3i} w_i'$ which adds three new vertices to the graph. Note that, since $|V(B)| = 2^{n-2}$ and $l = 2^{n-3}$, there are $2^{n-3} > r$ internal vertices that can be used as w_i's.

3. Map the constructed graph so that the roots of both copies of $B(n-2)$ are mapped to the single vertex- the root of the network.

Proposition 6 *Graph constructed in steps 1-2 by the above algorithm contains* $2^n + 1$ *vertices and* $2^n + 4r - 2$ *edges.*

Proof. Since $|V(B(n-2))| = 2^{n-2}$ the graph constructed in the first step contains $2 \cdot 2^{n-2} + 3 \cdot 2^{n-3}$ vertices. In the second step, we add $3r$ vertices which gives total of $2^n + 1$. Since we are connecting two trees by paths of length four, the number of edges after the first step is $2(2^{n-2} - 1) + 4 \cdot 2^{n-3} = 2^n - 2$. In the second step we add $4r$ edges which gives the total of $2^n + 4r - 2$. Note that $2^n + 4r - 2 = 2^n + \frac{2^{n-1}}{3} - \frac{2}{3}$.

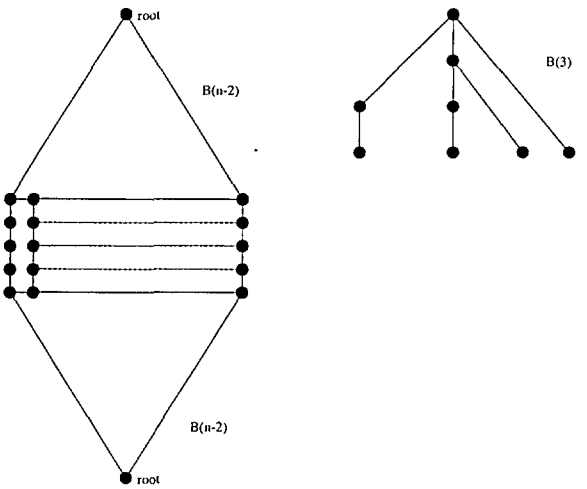


Figure 2: Construction of communication graph

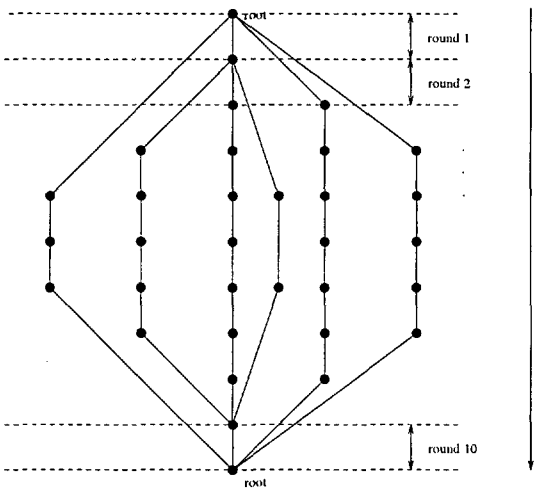


Figure 3: Polling in the K_{2^5}

Example 7 *If $N = 2^5$ then $r = 2$ and the communication graph is presented in Figure 3.*

4 Analysis for the n-cube

In this section we discuss polling primitive in hypercube networks. The n-cube Q_n is the graph (V, E) such that $V = \{(i_1, i_2 \ldots, i_n) : i_k \in \{0, 1\}\}$ and two vertices $(i_1, i_2 \ldots, i_n)$ and $(j_1, j_2 \ldots, j_n)$ are joined by an edge if and only if there is exactly one k such that i_k and j_k are different. The number of vertices in the n-cube is 2^n and the number of edges is $n2^{n-1}$. To improve the lower bound from previous section, we consider the following graph. Take K_{2^n-1} where $V(K_{2^n-1}) = \{1, \ldots, 2^n - 1\}$ and let $(x + K_{2^n-1})_n$ be the graph obtained from K_{2^n-1} by adding vertex x, the *root* and edges between x and i for $i = 1, \ldots, n$, i.e. with vertex set $V(K_{2^n-1}) \cup \{x\}$ and edge set $E(K_{2^n-1}) \cup \{\{x, 1\}, \{x, 2\}, \ldots, \{x, n\}\}$.

Proposition 8 *Let n be an even number. Polling in $2n$ rounds in $(x + K_{2^n-1})_n$ with the root x requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{12}\sqrt{2^n} - \frac{4}{3} \rceil$ messages.*

Proof. In the process of polling the algorithm constructs a layered graph with the kth level containing the vertices initiated in the kth round. It is convenient for our discussion to view the root x as a pair of vertices (x_1, x_2), where x_1 sends the messages, x_2 receives the messages. Denote by $G_1(l)$ a graph with vertex set consisting of levels 0 through l and all the edges in the communication graph between these levels, by $G_2(l)$ a graph with vertex set consisting of levels $2n - l$ through $2n$ and all the edges in the communication graph between these levels.

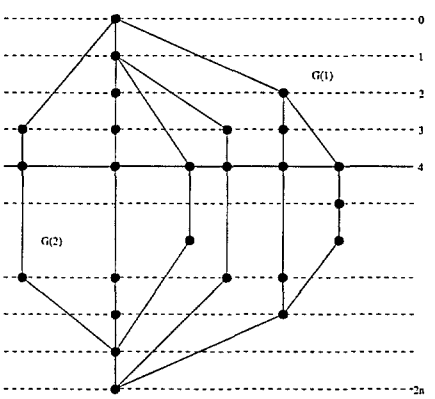


Figure 4: Construction of communication graph

Denote by h_1, (h_2) the height of G_1, (G_2) by d_1, (d_2) the degree of x_1, (x_2). Figure 4 illustrates a possible communication graph with $h_1 = 4$. Since the degree of the root is n we have, $d_1 + d_2 \leq n$. Let L_i denote the set of vertices on the ith level. Then the number of vertices in graph $G_1(n-2)$ is

$$|V(G_1(n-2))| = \sum_{i=0}^{n-2} |L_i|.$$

Similarly as in the proof of Proposition 3 we have $|L_0| = 1$ and

$$|L_i| \leq 2^{i-1} \tag{5}$$

for $i = 1, \ldots, d_1$ but since after round d_1, x_1 will not initiate any vertices we have

$$|L_i| \leq 2^{i-1} - 2^{i-d_1-1} \tag{6}$$

for $i = d_1 + 1, \ldots, n-2$. Therefore,

$$|V(G_1(n-2))| \leq \sum_{i=1}^{n-2} 2^{i-1} - \sum_{i=d_1+1}^{n-2} 2^{i-d_1-1} + 1$$

$$= 2^{n-2} - 2^{n-d_1-2} + 1. \tag{7}$$

Since the communication in G_2 can be viewed as reverse polling procedure, we also have

$$|V(G_2(n-2))| \leq 2^{n-2} - 2^{n-d_2-2} + 1. \quad (8)$$

The number of vertices in the communication graph is $2^n + 1$ and so

$$|V(G_1(n-2))| + |V(G_2(n-2))| +$$

$$|L_{n-1}| + |L_n| + |L_{n+1}| = 2^n + 1$$

which gives

$$|L_{n-1}| + |L_n| + |L_{n+1}| \geq 2^{n-1} + 2^{n-d_1-2} + 2^{n-d_1-2} - 1.$$

Therefore for some $n-1 \leq i \leq n+1$,

$$|L_i| \geq \frac{2^{n-1} + 2^{n-2}(2^{-d_1} + 2^{-d_2}) - 1}{3}. \quad (9)$$

Consider the function $f(d_1, d_2) = 2^{-d_1} + 2^{-d_2}$ subject to $d_1 + d_2 \leq n$. The function is minimized for $d_1 = d_2 = \frac{n}{2}$ and so we can further estimate the right hand side of (9)

$$|L_i| \geq \frac{2^{n-1} + 2^{n/2-1} - 1}{3}. \quad (10)$$

The number of messages sent is equal to the number of edges in the communication graph which by (10) is at least

$$2^n - 1 + |L_i| \geq 2^n + \frac{2^{n-1}}{3} + \frac{\sqrt{2^n}}{6} - \frac{4}{3}.$$

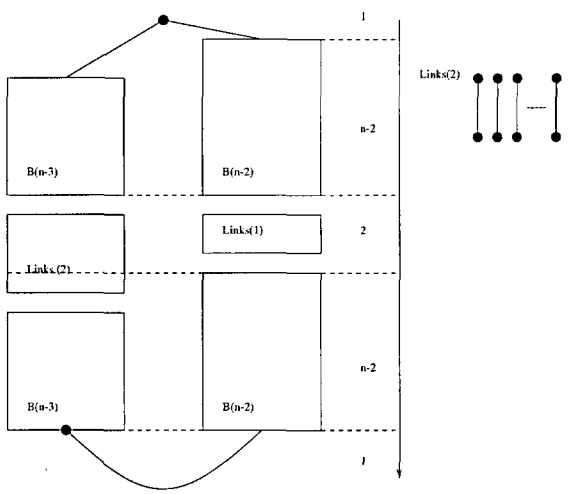


Figure 5: The communication graph for the n-cube

Since Q_n is a subgraph of $(x + K_{2^n-1})_n$ we have the following corollary.

Corollary 9 *Let n be an even number. Polling in $2n$ rounds in n-dimensional hypercube requires $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{6}\sqrt{2^n} - \frac{4}{3} \rceil$ messages.*

Next, we describe an algorithm that can be used to perform polling in an n-cube. In the communication graph, we will again make use of $B(n)$ trees described in Section 2. The algorithm uses the communication graph from Figure 5. Thus, the communication graph contains two special vertices x_1, x_2 which correspond to the root. Layer 1 contains one vertex which is the root of B_1, a copy the broadcast tree $B(n-2)$ (upper right box in Figure 5). Layer 2, in addition to some vertices of B_1, contains another vertex which is the root of B_2, a copy of the broadcast tree $B(n-3)$ (upper left box in Figure 5). The leaves of B_1 are connected by paths of length 2 (Links(1)) with the corresponding leaves of another copy of $B(n-2)$, B_1' (lower right box in Figure 5). The leaves of B_2 are connected to the corresponding leaves of the second copy of $B(n-3)$, B_2' (lower left box in Figure 5). The root of B_2' is x_2 and there is one more edge connecting the root of B_2' with x_2. Thus, technically x_2 is on level $2n+1$. The number of layers of the graph is $2n+1$ and since the broadcasting trees observe the polling requirements all the restrictions of the communication model are meat. It remains to show that the communication graph can be embedded into the n-cube. The n-cube embedding is illustrated in Figure 6. Figure 6 contains six boxes A, B, C, L, F, G which correspond to the boxes in Figure 5. Specifically:

- F consists of two layers: $F1$ contains $x \ldots x1110$, $F2$ contains $x \ldots x1010$.
- L is a broadcasting tree of height $n-3$ with leaves from $x \ldots x0110$, which can be easily constructed.
- G is a broadcasting tree of height $n-3$ with leaves from $x \ldots x1000$.
- A is a broadcasting tree of height $n-2$ with leaves from $x \ldots x011$.
- B has only one layer: $x \ldots x111$.
- C is a broadcasting tree of height $n-2$ with leaves from $x \ldots x101$

Note that the vertices in F_1 are connected to the corresponding vertices of F_2 as they differ in exactly one position (third position from right). Leaves of L can be connected to the leaves of G by paths of length 3 which look like $x \ldots x0110 - x \ldots x1110 - x \ldots x1010 - x \ldots x1000$. Similarly, leaves of A can be connected with leaves of C by paths of length 2 which look like $x \ldots x011 - x \ldots x111 - x \ldots x101$.

Example 10 *The concrete example of the embedding into 5-cube is illustrated in Figure 7.*

Proposition 11 *The algorithm terminates in $2n$ rounds and sends $2^n + 3 \cdot 2^{n-4} - 1$ messages.*

Proof. Tree $B(k)$ contains $2^k - 1$ edges and 2^{k-1} leaves, and so the total number of edges used is
$2(2^{n-2} - 1) + 2 \cdot 2^{n-3} + 2(2^{n-3} - 1) + 3 \cdot 2^{n-4} + 3$
$= 2^n + 3 \cdot 2^{n-4} - 1.$

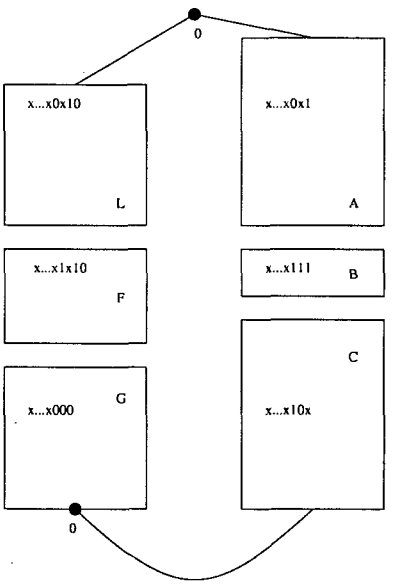


Figure 6: Embedding

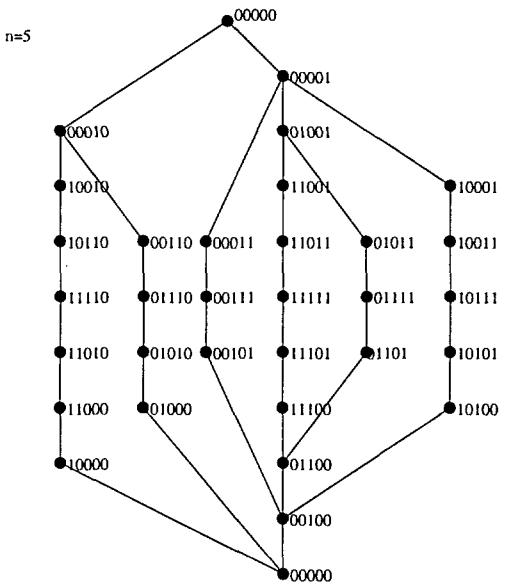


Figure 7: The communication scheme for 5-cube

5 Summary

We studied the polling problem in networks with complete graph and hypercube topologies. For a complete graph on 2^n vertices we showed that the polling primitive requires $2n$ rounds. We also showed that a polling algorithm in a complete graph K_{2^n} which terminates in $2n$ rounds must send at least $2^n + \frac{1}{3}2^{n-1} - 1$ messages. In addition, we presented an algorithm which completes polling in $2n$ rounds and sends the optimal number of messages.

For hypercube networks we established the lower bound of $\lceil 2^n + \frac{1}{3}2^{n-1} + \frac{1}{6}\sqrt{2^n} - \frac{4}{3} \rceil$ for the number of messages sent in $2n$ rounds by a polling algorithm and we presented an algorithm which uses $2^n + 3 \cdot 2^{n-4} - 1$ messages an completes polling in $2n$ rounds. Note that, unlike in the case of complete graphs, the lower and upper bounds are significantly different and it will be interesting to improve either of them.

Acknowledgments

Research partially supported by grant KBN 2 PO3A 023 09 and by NSF grant INT-9406971.

References

[1] D. Agarwal. E. Abbadi, "An Efficient Solution to the Distributed Mutual Exclusion Problem", Proc. 9th Symposium on the Principles of Distributed Computing, 1989.

[2] L. Bomans, D. Roose, "Benchmarking the iPSC/2 Hypercube Multiprocessor", Concurrency: Practice and Experience, Vol. 1, pp. 3-18, 1989.

[3] T. Chan and Y. Saad, "Multigrid Algorithms on the Hypercube Multiprocessors", IEEE Transactions on Computers, Vol. 35, pp. 969-977, 1986.

[4] A. Czygrinow, M. Karoński, V. S. Sunderam, "The Polling Primitive for Hypercube Networks", Proc. 7th IEEE Symposium on Parallel and Distributed Processing, 1995.

[5] T. H. Dunigan, "Performance of the Intel iP-SC/860 Hypercube", Oak Ridge National Laboratory, Technical Report TM-11491, 1990.

[6] J. Hromkovic, R. Klasing, B. Monien, R. Peine, "Dissemination of information in interconnection networks (Broadcasting & Gossiping)", *Combinatorial Network Theory* (D. Du and D. Hsu Eds.), pp. 125-212, Kluwer Academic Publishers 1990.

[7] Y. Lan, A. Esfahanian, and L. M. Ni, "Multicasting in Hypercube Multiprocessors", Journal of Parallel and Distributed Computing, Vol 8, pp. 30-41, 1990.

[8] A. Rescigno, "On the communication complexity of polling", Information Processing Letters 59, pp. 317-323, 1996.

[9] A. Rescigno, "Optimal Polling in Communication Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 8, No. 5, pp. 449- 461, 1997.

[10] Y. Saad and M. Schultz, "Topological Properties of Hypercubes", IEEE Transactions on Computers, Vol. 37, pp. 867-872. 1988.

Strategic IS Planning From the Slovenian Business Perspective

Andrej Kovačič, Aleš Groznik, Jurij Jaklič, Mojca Indihar Štemberger,
Talib Damij, Janez Grad, Miro Gradišar and Tomaž Turk
University of Ljubljana, Faculty of Economics,
Kardeljeva ploščad 17, SI-1000 Ljubljana, Slovenia
Tel: +386 61 1892 400, Fax: +386 61 1892 698
E-mail: andrej.kovacic@uni-lj.si

Keywords: strategic information system planning, business information systems, survey, questionnaire

Edited by: Rudi Murn
Received: October 29, 1999 **Revised:** April 6, 2000 **Accepted:** April 27, 2000

In recent years there has been a dramatic change in business environment resulting in reengineering of key business activities and processes. Among others, the role of information system (IS) has significantly increased as organisations have employed information technology (IT) to improve the capture, processing and distribution of information. Information became an important asset to the company, which is carefully monitored, planned and upraised. The paper presents the results of a survey on the strategic IS planning practices of Slovene companies. It highlights the participation, critical success factors and main benefits of strategic IS planning. The results show that comparing to similar studies very low number (50%) of responding companies were performing strategic IS planning. It is also interesting that in Slovene companies the leading initiator is top management (36%) whereas the role of IS management is surprisingly modest (23%). Main benefits of strategic IS planning from the Slovene business perspective are improved internal co-ordination, efficient and effective management of IS resources and improved productivity.

1 Introduction

There are a number of researches focused on identifying key IT issues concerning corporate transformation. Technical progress together with the opening of a global market is definitely among the primary factors playing roles in modern society. IT is an essential component of a firm's strategy in a global market. One of the consequences of recent development in the field of information technology is an ongoing process of planning in both the IS and business arenas.

Slovene organizations react very differently to projects or attempts at introducing modern IT and renovation of business processes, though the purpose is clear: reduction of costs, shortening the business cycle, and improvement of quality. The difficulties in the public sector are larger than those in the private sector. The increased employment in the public sector during the past few years has further entrenched bureaucracies; the problems of efficiency are then most often solved through purchasing computer hardware and software. Moreover, if managers feel the corporation they work for is successful at the present time, they usually reject the idea of strategic IS planning and renovating the business. Of course, when a company faces trouble, there never seem to be enough financial or human resources to start such a project. Coping with these problems while working on IS renovation projects in the last few years, we have noticed (Kovačič, 1999) that IT plays the key role in business process renovation and a strong correlation between the quality of IS within an organization, and improvement of overall corporate culture and strategies (Lederer, Sethi, 1996). We must also keep in mind that an incorrect or inadequate strategic IS planning can deliver partial solutions which do not consider the system as a whole and are by all means unsatisfactory.

Strategic IS planning is the process of identifying a portfolio of computer-based applications that assists an organisation in executing its business plans and realising its business goals (Lederer, Salmela, 1996). Although the importance of strategic IS planning is clearly identified (Karimi, Gupta, Somers, 1996), (Lederer, Sethi, 1996), (Lederer, Salmela, 1996), (Porter, 1985) practical experience on strategic planning is very scarce. The lack of information encouraged us to perform a systematic analysis of strategic IS planning practices in Slovenia.

The paper presents the results of a survey on the strategic IS planning practices of Slovene companies. It highlights the participation, critical success factors and main benefits of strategic IS planning. The results are compared to the results of similar studies (Pavri, Ang, 1995), (Teo, Ang, Pavri, 1997). Different place and time of that investigation had to be considered.

2 Methodology

The purpose of the study was to analyse the strategic IS planning practices in Slovenia. The study was performed

by the MIS department of the faculty of Economics in Ljubljana in 1998 and was based on a questionnaire (can be optained from http://www.ef.uni-lj.si/projekti/informatika) that was previously developed by Teo, Pavri and Ang (Pavri, Ang, 1995), (Teo, Ang, Pavri, 1997). We found the coverage of the questionnaire a very good basis for evaluation of strategic IS planning situation and was therefore left unchanged in order to made the comparison of the results between present and Teo et al's study feasible. The questionnaire was sent to IS executives in several Slovene organizations which were asked to provide information by answering the questions on the following subjects: organization of the MIS departments, the state of IS, the use of new concepts and technologies in the development of IS, databases, data warehouses and IS strategic planning.

The answers to the first section provided general information about the company, its structure and general state of the IS, second part provided information about the architecture of IS and the underlying technology. The focus of the third section was the state of databases and data warehouses and the last part of the questionnaire investigated IS strategic planning. We are planning to repeat the survey every two years which will help us compare the results and observe current trends in Slovene organizations over a longer timeframe.

After eliminating the missing and illogical answers, we got the total number of answers to all the four parts of the questionnaire (181 to the first part, 175 to the second part, 166 to the third and 131 to the last part). Table 1 shows the structure of the organizations according to its activities. The activities in the category other is of a different kind such us consulting, transport, IT, catering, tourism, health service, government, telecommunications.

This paper focuses only on the IS strategic planning part of the questionnaire which covers the following topics:

- The participation in strategic IS planning
- The strategic IS planning critical success factors
- The benefits of strategic IS planning
- Company and MIS department degree of maturity
- Other relevant IS planning data (e.g. planning methodology, corporate and IS plans alignment).

3 Results

The study involved 450 large Slovene companies from a wide range of industries. The size of the companies was defined according to the number of employees and the revenues in 1997 (Slovene Corporate Law, 1993). A company classified as large when met both criteria: more than 250 employees and the revenues over 4 million USD. A total of 131 useful returns to the IS planning part were obtained, representing the database on strategic IS planning practices in Slovenia. The rate of the return was 29% and is comparable with the similar studies (Karimi, Gupta, Somers, 1996), (Lederer, Sethi, 1996), (Pavri, Ang, 1995), (Teo, Ang, Pavri, 1997), (Torkzadeh, Xia, 1992) conducted in the past where the rate of the return reached 21%, 24%, 22%, 20% and 23% respectivelly. Considering the length (21 pages) and complexity of open and closed questions, the number of useful returns is quite encouraging and is showing that strategic IS planning is becoming more and more important in Slovenia.

Analysis of the returned questionnaires shows that 66 (over 50%) of the responding companies were performing some form of IS planning process. As can be seen from Table 2, the relationship in Teo et all's study was better since 63% of companies have implemented some form of IS planning process. This is especially worrying since Teo et al's study was performed two years earlier.

Since we are planning to repeat this study every two years it is going to be very interesting to observe how the IS strategic planning process in Slovenia will develop.

3.1 IS strategic plan/corporate plan

As has already been presented in the past, the key to the success of the strategic IS planning process is in corporate and strategic IS plan alignment (Clarke, 1992), (Lederer, Sethi, 1996), (Lederer, Salmela, 1996). Although the rate of companies conducting the strategic IS planning in Slovenia is much lower that the one in Singapore, it is surprising that corporate and strategic IS plans are aligned in much higher rate (92.4% compared to Teo et al's 79.3%) as shown on Table 3.

Furthermore, in majority of companies (96.9% in the present study, 93.1% Teo et al's), the IS strategic plan developers show a high level of awareness of corporate objectives. Such results suggest that those companies that perform strategic IS planning realise the importance of corporate and strategic IS plan alignment as the key to the successful role of IT in business environment.

3.2 Planning methodologies/participants in IS planning

Table 4 shows the comparison of the planning methodologies used by companies. Of the 65 respondents, 39 (60% compared to Teo et al's 69%) stated that they used a combination of top-bottom and bottom-up planning methodologies. In addition, 24 respondents (36.9% compared to Teo et al's 19%) used top-down planning approach. This result suggests that combination of methodologies prevails as most usual IS planning methodology, enabling synergy of business and user involvement. Dissimilar to Teo et al's finding our results suggest that top-down approach is more widely used in Slovenia. This indicates that IS planning in Slovenia is still traditionally oriented process in which management plays very important role. Table 5 shows the portfolio of participants involved in strategic IS planning in which top and MIS management plays the predominant role (on the scale from 1 to 5, 2.92 and 2.43 respectively

Business activity	Number	Percentage
Manufacturing	75	41%
Commerce	31	17%
Finance and insurance	13	7%
Mixed	7	4%
Other	55	30%

Table 1: Structure of organizations based on business activity

IS strategic plan	Present study			Teo et al.		
	Number	Percentage	Rank	Number	Percentage	Rank
Existing	66	50.4%	1	58	63.0%	1
Non-existing	65	49.6%	2	34	37.0%	2

Table 2: IS strategic plan implementation

in the present study compared to Teo et al's 3.79 and 3.36) comparing to users involvement (1.55 compared to Teo et al's 2.8).

Comparing strategic IS planning methodologies and participants we can conclude that combination of both top-bottom and bottom-up planning methodologies is still prevailing. High involvement of top and MIS management and significant lack of users participation results in high rate of top-down approach.

3.3 Critical success factors

Among 10 critical success factors listed, first 4 in our study were related to importance of management involvement and support as well as human resources related issues.

Getting top management support for the planning efforts (4.83 in the present study, 4.69 Teo et al) with having a clear-cut corporate plan guide IS planning efforts (4.52 in the present study, 4.41 Teo et al) represent the key drivers for successful strategic IS planning in the literature (Clarke, 1992), (Karimi, Gupta, Somers, 1996), (Lederer, Sethi, 1996).

The ability to obtain sufficient qualified personnel ranks as the second most important critical success factor in strategical IS planning personnel in Slovenia (4.59 in the present study, 4.22 Teo et al). We believe that the reason for this deviation is a significant shortage of qualified resources to support increasing evolution and spread of information technology.

The fourth most important critical success factor is good user-IS relationships (4.38 in the present study, 4.22 Teo et al). This relationship is crucial for achieving the strategic objectives. Users and IS staff should act as partners in meeting the strategic objectives which would lead the company to operational excellence as already proved in the past (Karimi, Gupta, Somers, 1996), (Lederer, Sethi, 1996), (Lederer, Salmela, 1996), (Porter, 1985). Other success factors (see table 6) are mainly planning related (i.e. time management, environmental changes, planning procedure, etc).

3.4 Benefits/satisfaction with strategic IS plan

According to the results shown in Table 7, companies highly appreciate the benefits from strategic IS planning process (on a scale from 1 to 5, all benefits were rated with a mean of 3.89 or higher). The most important benefits were in both studies improved internal co-ordination (4.57 in the present study, 4.07 Teo et al), efficient and effective management of IS resources (4.45 in the present study, 4.05 Teo et al) and improved productivity (4.37 in the present study, 4.09 Teo et al). It is also interesting to observe that respondents value internal benefits more than external. The possible reason is that internal benefits are easily recognised whereas external are not clearly defined.

A comparison of the two studies reveals that although the most important benefits match, they were ranked differently. Improved productivity, which was the most important benefit in Teo et al's study ranked only third in the present study. This is somehow interesting since improved productivity is historically the most important benefit from strategic IS planning process (Davenport, Linder, 1994), (Karimi, Gupta, Somers, 1996), (Lederer, Salmela, 1996), (Porter, 1985), (Torkzadeh, Xia, 1992), that was in the present study clearly underscored (10 respondents rated improved productivity below semi-beneficial). This indicates that many Slovene executives perhaps still do not understand the strategic role and benefits of IS. The change of traditional thinking using IS for internal co-ordination and efficient and effective management support will have to be changed to improve productivity as well as external benefits. The first step to achieve this change should be a part of strategic IS planning process.

IS strategic plan allignment	Present study			Teo et al.		
	Number	Percentage	Rank	Number	Percentage	Rank
Alligned	61	92.4%	1	46	79.3%	1
Not-alligned	2	3.0%	3	12	20.7%	2
Corporate plan non-existing	3	4.5%	2	0	0.0%	3

Table 3: Corporate and strategic IS plan alignment

Planning methodologies	Present study			Teo et al.		
	Number	Percentage	Rank	Number	Percentage	Rank
Bottom-up	2	3.1%	3	7	12.1%	3
Top-down	24	36.9%	2	11	19.0%	2
Combination of above	39	60.0%	1	40	69.0%	1
No answer	0	0.0%	4	0	0.0%	4
Total	65	100.0%		58	100.0%	

Table 4: IS planning methodologies

Similarly to highly appreciated benefits from strategic IS planning process, the satisfaction with strategic IS plan also ranked high. Over 98 percent of respondents rated the satisfaction with their strategic plan above average.

3.5 Initiation of strategic IS planning process

The results presented in Table 8 show that initiators of strategic IS planning process vary significantly between Slovenia and Singapore. Whereas Teo at al's study shows natural rank of initiators (41.4% IS management; 25.9% top, IS and line management; 12.1% top and IS management), present study reveals that in Slovenia the most important initiator of strategic IS planning process is top management (35.9%), followed by top and IS management (28.1%) and IS management (23.4%).

The responses regarding the initiation of strategic IS planning process confirmed that in Slovenia IS planning is still traditionally oriented process in which top management plays very important role. This is very surprising since we would expect IS management to significantly add value to the strategic IS planning due to it's expertise.

It is also very revealing to note that top, IS and line management does not take joint initiation in Slovenian companies. In fact, joint management initiation rated last with only 3.1% in contrast with 25.9% in Teo et al's study.

3.6 Evaluating IS function

It is interesting to observe that among 66 companies that practised strategic IS planning process, only 15 (23.1%) have objective measures of IS contributions to productivity, although 95% of respondents rated the importance of developing such measures with 3 or higher on a scale from 1 to 5. This result is in line with Teo et al's result that shows 24% respondents have objective measures and 98% rated the importance 3 or higher.

The lack of objective measures of IS contributions to productivity is also connected to benefits from strategic IS planning. Since very few respondents have objective measures of IS contributions to productivity there is no mechanism to measure the impact on productivity of business processes resulting in poor rating of improved productivity as a benefit from strategic IS planning (Table 7).

Nevertheless, results show that the importance is recognised by the companies, but not yet implemented in practise. It is going to be very interesting to see how this subject is going to develop in the future since the impact of information technology on productivity remains an important benefit.

3.7 Company's degree of maturity

The company's degree of maturity has been evaluated through long range business planning, capital allocation and objective setting. It is very encouraging that 84.6% of responding companies perform long range business planning in either more tactical than strategic (47.7%) or clearly strategic nature (36.9%).

Financial aspect of the capital allocation is significant since 95.4% of respondents have set capital allocation criteria. Out of these 64.6% of respondents perform rigorous financial analysis with (33.8%) or without (30.8%) post audit. This results show that most companies perform serious financial analysis as a part of their planning process.

In case of clear objective setting, our study shows that the majority of respondents set the objectives (93.8%). It is interesting that there is a spread of only 12.3% between the top three objective settings; highly targeted in-

Participants (scale from 0 to 3)	Present study				Teo et al.		
	Number	Mean	S.D.	Rank	Mean	S.D.	Rank
MIS managers	64	2.92	0.32	1	3.79	0.59	1
Top managers	65	2.43	0.76	2	3.36	0.74	2
System analysts (developers)	55	2.20	0.86	3	2.75	0.88	4
Non-MIS managers	63	2.02	0.85	4	2.58	0.84	5
Consultants	62	1.92	1.05	5	1.89	0.92	9
Computer systems programmer	62	1.77	0.99	6	2.29	1.02	6
Computer operations personnel	57	1.56	0.94	7	2.05	0.91	7
Users	64	1.55	0.90	8	2.80	0.80	3
Vendors	58	1.36	1.03	9	2.02	0.86	8

Table 5: Participants in IS planning

dividual objectives with strong follow-up directly affecting compensation (32.3%), only generalised individual objectives (24.6%) and highly targeted individual objectives with strong follow-up (20%). This indicates that there is a wide variety of objective setting practises currently in place in Slovene companies leading us to a conclusion that this area should be better focused on and improved in the future.

3.8 MIS department's state of maturity

The MIS department's state of maturity has been evaluated through computer operations, system development, user involvement and feasibility assessment. The study shows that 14.8% of respondents stated that users are dissatisfied with the timeline and accuracy of computer operations. That is clearly a number that is not to be overlooked and computer operations should be the area where MIS departments must improve. One reason for dissatisfaction could be that extent of users participating in strategic IS planning is very low (1.55 in Table 5) although one of the most important critical factors in strategic IS planning is good user-IS relationships (4.38 in Table 6).

System development and users involvement parts of the questionnaire are possibly the most questionable in terms of data quality since our respondents are from the IS department. The study nevertheless shows that in the majority of respondents users are very confident of the MIS group's ability to consistently deliver major systems approximately on time, within budget and meeting specifications (66.2%) as well that users are involved only as much as necessary to define the system specifications and to implement it (63.1%), we must not forget that the questionnaire was filled in by IS executives. Ratings of IS department on MIS performance and users involvement might therefore be biased.

In case of feasibility assessment it is very surprising to note that in 24.6% of companies no formal standard for assessing the feasibility of proposed major systems development projects exists. Feasibility studies are nevertheless performed in 75.4% of companies. This share should increase in the future since resource management is becoming of vital importance in today's business environment.

4 Conclusions

Although the importance of strategic IS planning is clearly identified, the study shows that a moderate number (50.4%) of Slovene companies are involved in strategic IS planning. This is surprisingly low if we take into consideration that Teo et al's study reveals 63% of Singapore companies were performing strategic IS planning in 1996.

On the other hand, it is encouraging that those companies that perform strategic IS planning have corporate and strategic plans aligned (92.4%), enabling them to meet overall business plans and goals. The study also shows that strategic IS planning in Slovenia is still traditionally oriented process in which top management plays an important role since top managers are the key initiators and participants in IS planning in which top-down approach is broadly used. This is very surprising since we would expect the role of IS management to be significant due to their expert knowledge and experience. On top of the sometimes diminished role of IS management, the study shows that 14.8% of respondents stated users are dissatisfied with the timeline and accuracy of computer operations. This indicates that apart from underperforming role of IS management, users involvement in a strategic IS planning is also insufficient (users participation ranked last in the present study), although good user-IS relationship is one of the key success factors in strategic IS planning.

Overall strategic IS planning process is still one of the key business activities where Slovene companies will have to improve in order to be able to effectively participate on the overall global market of the information era.

References

[1] R. Clarke: Strategic Information Systems: Retrospect and Prospect, *International Conference on Information Systems and Organisations*, Bled, Slovenia, 1992.

Critical success factors (scale from 0 to 5)	Present study				Teo et al.		
	Number	Mean	S.D.	Rank	Mean	S.D.	Rank
Getting top management support for the planning efforts	64	4.83	0.38	1	4.69	0.54	1
Being able to obtain sufficiently qualified personell to do a proper job	64	4.59	0.58	2	4.22	0.75	3
Having a clear-cut corporate plan to guide IS planning efforts	64	4.52	0.73	3	4.41	0.80	2
Good user-IS relationships	64	4.38	0.76	4	4.22	0.68	3
Investing sufficient 'front end' time to ensure that all planning tasks and individiual responsibilities are well understood	64	4.31	0.73	5	3.98	0.78	8
Anticipating likely changes in information technology (and environmental changes) which might affect the strategic IS planning process	64	4.28	0.74	6	4.10	0.67	5
Having free comunication and commitment to change throught the organisation	63	4.21	0.91	7	4.02	0.81	7
Having a clear, concise, formal, planning procedure	64	3.98	0.86	8	4.05	0.85	6
Deciding on an appropriate planning horizon	64	3.89	0.89	9	3.95	0.60	9
Taking into account the people and politics side of strategic IS planning system	63	3.65	1.09	10	3.55	0.82	10

Table 6: Critical success factors in IS planning

[2] T. Davenport, J. Linder: Information Management Infrastructure: the new competitive weapon? *Proceedings of the 27th Hawaii International Conference on System Sciences*, 1994, 885 - 896.

[3] J. Karimi, Y. P. Gupta, T. M. Somers: Impact of Competitive Strategy and Information Technology Maturity on Firm's Strategic Response to Globalisation, *Journal of MIS*, 12(1996), 55 - 88.

[4] A. Kovačič: Information Technology as an Enabler to Enterprises in Transition, *Proceedings of The Third International Conference 'Enterprise in Transition'*, University of Split, Split-Šibenik 1999, 278 - 290.

[5] . L. Lederer, V. Sethi: Key Prescriptions for Strategic Information Systems Planning, *Journal of MIS*, 13(1996), 35 - 62.

[6] A. L. Lederer, H. Salmela: Toward a Theory of Strategic Information Systems Planning, *Journal of Strategic Information Systems*, (1996), 237 - 253.

[7] F. N. Pavri, J. S. K. Ang: A study of the strategic planning practices in Singapore, *Information and Management*, 28(1995), 33 - 47.

[8] M. E. Porter: Competitive Advantage: creating and sustaining superior performance, Free Press, New York, 1985.

[9] T. S. H. Teo, J. S. K. Ang, F.N. Pavri: The state of strategic IS planning practices in Singapore, *Information and Management*, 33(1997), 13 - 23.

[10] G. Torkzadeh, W. Xia: Managing Telecommunications by Steering Committee, *MIS Quarterly*, 16(1992), 187 - 199.

[11] Zakon o gospodarskih družbah (Slovene Corporate Law), Uradni list RS-30/93(1993), Ljubljana.

Benefits from strategic IS planning process (scale from 0 to 5)	Present study				Teo et al.		
	Number	Mean	S.D.	Rank	Mean	S.D.	Rank
Improved internal coordination	65	4.57	0.63	1	4.07	0.71	2
Efficient and effective management of IS resources	65	4.45	0.66	2	4.05	0.59	3
Improved productivity	65	4.37	0.81	3	4.09	0.64	1
Improved quality in products/services	65	4.23	0.87	4	3.88	0.69	6
Improved competitive position	64	4.16	0.91	5	4.00	0.81	4
Sound technology path and policies	65	3.97	0.80	6	3.70	0.63	7
Larger market share	63	3.90	1.00	7	3.30	0.91	8
Greater ability to meet changes in the industry	62	3.89	1.11	8	3.89	0.76	5

Table 7: Benefits from strategic IS planning process

Initiated by	Present study			Teo et al.		
	Number	Percentage	Rank	Number	Percentage	Rank
Top management	23	35.9%	1	5	8.6%	4
Top and IS management	18	28.1%	2	7	12.1%	3
IS management	15	23.4%	3	24	41.4%	1
IS and line management	3	4.7%	4	3	5.2%	6
Line (or functional) management	2	3.1%	5	0	0.0%	7
Top, IS and line management	2	3.1%	5	15	25.9%	2
Missing data	1	1.6%	7	4	6.9%	5

Table 8: Initiators of strategic IS planning process

Degree of maturity	Number	Percentage
Long range business planning		
No formal long-range business plan	3	4.6%
Mostly financial and headcount projections	7	10.8%
More tactical than strategic	31	47.7%
Clearly strategic in nature	24	36.9%
Capital allocation		
No formal capital allocation criteria	3	4.6%
Formal document stating purpose and level of investment, but no financial measure of attractiveness	20	30.8%
Rigorous financial analysis for all major expenditures but no post audit	20	30.8%
Rigorous financial analysis with post audit	22	33.8%
Objective setting		
No formal setting of individual objectives	4	6.2%
Only generalized individual objective are set	16	24.6%
Highly targeted individual objectives are set but no formal follow-up or appraisal of results	11	16.9%
Highly targeted individual objectives with strong follow-up	13	20.0%
Highly targeted individual objectives with strong follow-up directly affecting compensation	21	32.3%

Table 9: Company's degree of maturity

Stage of maturity	Number	Percentage
Computer operations		
Users are dissatisfied with the timeline and accuracy of computer operations	9	14.8%
Users are generally satisfied with timelines and accuracy of computer operations but no formal production statistics are communicated to them	26	42.6%
Production control has been formalized, production objectives are set and performance versus plan is communicated to users on a regular basis	26	42.6%
Systems development		
No formal standard for systems development exists	8	12.3%
Users have little confidence in the MIS group's ability to deliver major systems on time, within budget and meeting specifications	14	21.5%
Users are very confident of the MIS group's ability to consistently deliver major systems approximately on time, within budget and meeting specifications	43	66.2%
Users Involvement		
Users are rarely involved in the systems development process	2	3.1%
Users are involved only as much as necessary to define the system specifications and to implement it	41	63.1%
Users are actively involved in all phases of the system development process and often manage the project team	22	33.8%
Feasibility assessment		
No formal standard for assessing the feasibility of proposed major systems development project exists	16	24.6%
Feasibility assessments are well defined and required for all proposed major system development project but no post-implementation audit	15	23.1%
Feasibility assessments are well defined and required for all proposed major system development projects and followed by post-implementation audits	34	52.3%

Table 10: MIS department's state of maturity

Application Modeling and Concurrency Control in Active DBMS: A Survey

Prithwish Kangsabanik, R. Mall and A.K. Majumdar
Dept. of Computer Science and Engg.
I.I.T Kharagpur, India. PIN - 721 302.
email: {prith, rajib, akmj}@cse.iitkgp.ernet.in

Keywords: Active database, Concurrency control, Object oriented systems.

Edited by: Rudi Murn
Received: October 5, 2000 **Revised:** March 5, 2000 **Accepted:** April 19, 2000

In this paper, we survey some of the recent developments in application modeling and concurrency control in Active DBMS (ADBMS). We first review the evolution of Active DBMS and different application areas of ADBMSs. Application modeling techniques for ADBMS applications have been surveyed after that. Then we discuss about the execution modeling and concurrency control in active DBMS – which is one of the most challenging areas in active database research. Several important active database research projects are also reviewed with mention of their contributions and current research directions pointed out by these projects.

1 Introduction

Databases are traditionally seen as passive repositories of facts. Actions performed on the database are insertion, rearrangement, modification and retrieval of data. The role played by a database is similar to that played by a bookshelf. We do not expect the bookshelf to reject an unsuitable book. Similarly we do not expect databases to do autonomous work, such as actively constraining their contents, based on environmental cues. This passive view of databases affects the way we think of interacting with them, the jobs for which we use them, and the way we design systems containing them.

This traditional perspective is insufficient for many applications such as computer integrated manufacturing (CIM), office workflow control, process control, stock control, battle management, network management etc., which require timely response to critical situations. For *time-constrained* applications, it is important to monitor conditions defined on the states of the database, and then once these conditions are satisfied, to invoke specified actions, subject to some timing constraints. For example, inventory control in an automated factory may require that the available quantity (stock) of each item be monitored; if the stock for some item falls below a threshold, then a reorder procedure has to be initiated before the end of one working day. In a *situation assessment application* which requires various targets to be tracked, if a target is discovered within a critical distance, then an alert message may have to be displayed on the commander's screen with the highest priority. The passive DBMSs are not equipped to deal with applications requiring such autonomous actions.

An *Active DBMS* [62] (ADBMS hereafter) is a database system that incorporates user-definable responsive components which can autonomously execute actions affecting both the database and the external environment. It attempts to ensure both modularity and timely response. To achieve these objectives, often *event-driven production rules* are incorporated into the databases. These rules are triggered by database operations and perform suitable actions. Situations, actions and timing requirements are all specified declaratively to the system using the rules. The system monitors the database state, triggers appropriate actions when certain situations become true, and schedules tasks to meet the timing requirements without user or application intervention.

1.1 Evolution of Active DBMSs

Initial attempts to unite rule functionalities with the databases involved interfacing databases and expert systems [192]. Such an *interfacing approach*, while useful in some applications, has encountered several major problems:

- *The interface is a barrier.* Since expert system rules are separated from the database, event detection of database events and rule optimization is much more difficult.
- *Time cost of interface.* The interface can introduce a significant performance penalty in time.
- *Data structure mismatch.* Expert system rules are instance oriented but database languages are set oriented.
- *Rule execution mismatch.* Expert system rules often execute using either forward or backward chaining, event driven rules typically execute using forward chaining.

An approach to overcoming these problems is to integrate rule functionalities fully into the database itself, resulting in an *Active DBMS*. Rules are defined using database languages, and are executed by the database runtime system. Such an *integrated approach* is highly desirable, since it removes the interface barrier and its cost. Full integration, implies harmonizing rule functionality with the services already promised by the database to its users. These services include:

- Persistence of data between executions.
- Safe concurrent access to data.
- Data independence.
- Efficient access through multiple indices.
- Uniform associative access language.
- Protection against hard and soft failures.
- Schematic organization.
- Logically centralized data.
- Security.
- Automated integrity enforcement.

In an active database, each database service must be carefully reconsidered, and perhaps redefined. For example,

- Transactions containing rules can spawn and signal subtransactions by triggering rules. Streams of long running inter-communicating hierarchically structured transactions could be constructed this way. This suggests the need for a new transaction framework for handling communicating transactions.
- User transactions should not see data inconsistencies due to concurrently executing rule actions. Rule actions must be integrated into the concurrency control framework of the database.
- Rule actions (trigger expressions, predicates) must be integrated with existing database languages, so the user sees a single mode of data access.

Although the rules and rule processing have been extensively investigated in the expert systems and deductive database literature, there are some important differences in the semantics of rules in active databases and steps to be taken for rule processing in such an environment. Some of these issues are discussed below.

1. The rules in active databases are event driven and are typically invoked by operations or state changes caused by a transaction.
2. Multiple rules may be eligible for execution and hence their (concurrent) execution needs to be consistent with the semantics used for transaction execution.
3. As active capability is intended as a uniform mechanism for supporting several database functionalities (e.g. integrity constraints, view materialization), the rules are treated as any other shared data and are managed by the system.

Based on application requirements, the events to be monitored can be categorized into the following:

- *Database events :* typically insert, delete, and modify operations in a relational DBMS or method invocations in an object-oriented DBMS.
- *Temporal events :* typically absolute, relative events.
- *Abstract or external events :* events that are detected outside the scope of the DBMS; the condition-action portion of the rule is processed by the DBMS.

Active capabilities in DBMSs can be traced back to the ON conditions of CODASYL [1]. Triggers were proposed in System R [78, 79] as a mechanism for enforcing integrity constraints (or "assertions"). The use of triggers for maintaining materialized views, snapshots, derived attribute values, and some algorithms for implementing them have been described in [34, 141, 168, 155, 22, 125, 151, 113]. The term "active database" was coined in [155] to describe a system that supports automatic updates of views and derived data as base data are updated. Simple triggers (where the triggering conditions involve only a single relation) are supported by some current commercial relational DBMSs (e.g. [61]). Time triggers, where the triggering condition is a point in time (e.g. at 2:00:00 on 5/2/1998), have been used for office system applications in [205, 15]. In [80, 183], Stonebraker et. al. have pointed out the utility of production rules (i.e. situation-action) as a unifying mechanism for integrity control, access control and view processing, and for supporting inference via forward and backward chaining. Of course, the AI community has long been using production rules [85], actors [123], daemons, active objects [23], and procedural attachment of slots of frames [154, 23] for "active" knowledge representation and inference mechanisms. However, these representations and their implementations assume small number of objects (rules, facts) stored in the main (or virtual) memory, not in a large database on secondary storage. Also, they typically assume a single thread of execution, and hence do not provide any concurrency control over shared objects (as DBMSs do).

In recent years several active database systems have been designed and some have been implemented. Three early active database systems are Ariel [119], the second version of the POSTGRES rule system (PRS-II) [188], and Chimera [39, 40, 86]. Ariel has a rule language and execution semantics based closely on the expert system OPS5 [28], and incorporates a production rule language originally designed for expert systems. The Ariel project has focused on the design of an OPS5-like rule language for database setting, and on methods for highly efficient rule

condition testing using a variations of *Rete* and TREAT algorithms designed for OPS5 [196]. The Ariel rule language is fully implemented using Exodus database toolkit [119]. The POSTGRES rule system, sometimes referred to as PRS-II to distinguish it from an earlier proposal [186], focuses on both the language and implementation of several different classes of rules, each appropriate for a particular suite of applications. The Chimera system combines object-oriented, deductive, and active database technology. The first prototype of Chimera has also been implemented, employing some techniques adapted from Starburst [39]. Two other relational active database projects are DATEX [25] and DIPS [175]. These projects implement OPS5 rule language using an underlying database system and special indexing technique to support efficient processing of large rules and data sets. The PARADISER project also uses a database system for efficient processing of expert system rules. PARADISER, in addition, focused on distributed and parallel rule processing [68]. RPL (for Relational Production Language) was an early project in relational active database system. RPL includes an OPS5-like rule language based on relational queries. In a prototype implementation of RPL rule processing is loosely coupled to a commercial relational DBMS [67]. The *Alert* project explores how active rules can be supported on top of a passive database system with minimal extensions [174]. Finally, Heraclitus [108] is a relational database programming language with *delta relations* as first-class objects; a primary goal of the Heraclitus language is to simulate and support active rule processing [108].

Beginning with an early project HiPAC [63] several recent efforts (including Chimera) have considered active object-oriented database systems. HiPAC was a pioneering project in the area of active object oriented database systems. HiPAC includes a very powerful rule language for an object oriented data model, a flexible execution semantics, and several main-memory experimental prototypes [63]. Other recent active object oriented database projects are Ode [6, 99], Sentinel [11, 47], REACH [32, 24, 31], SAMOS [98, 97], ADAM [71, 69].

Though "active functionality" has been claimed for many database systems, it should be clear which functionalities a database management should support in order to legitimately considered as an active database management system. Thus, in [72], a rulebase of active database features has been provided. Here, it has been distinguished between mandatory features that are needed to qualify as an active database system, and desired features which are nice to have. Also, in [72], a set of application classes for ADBMS applications and the corresponding subset of functionalities needed for each of these application class has been identified.

In this paper, several important research areas in active database are surveyed. We begin with a brief discussion on some of the important application areas of active databases, e.g. constraint and integrity management, workflow management, cooperative problem solving etc. One important research area lacking focus, in spite of the apparent popularity of ADBMSs, is the development of a suitable application modeling techniques for active DBMS. We review some important works on application modeling on passive DBMS first. Then we discuss how these works have been transformed with the advent of object oriented modeling, which gives better modeling power in the presence of changing requirements, which are to a large extent due to the changes in the business environment. However, it has been realized that a more intuitive way to describe business policies is in terms of rules, which is the basic backbone of active DBMS. New modeling techniques which are coming up for developing active DBMS applications have been discussed.

One of the important areas of research in ADBMS is the execution modeling of active DBMS transactions. The different extended transaction models have been used for ADBMS transaction modeling and execution modeling. Variation of the existing extended transaction models have been used to develop the concurrency control algorithms for event driven execution of ADBMS transactions. We give an extensive review of the execution model and concurrency control algorithms for ADBMS transactions in this paper.

To make this paper complete, we discuss the major projects that have been carried out in ADBMS and current research directions investigated by these projects. Several prototype active database systems are being or have already been developed. The underlying data models used in these databases are mostly relational or object oriented. To capture the execution semantics, most of these systems have adopted the rule based approach. We first discuss relational active database projects and the research directions investigated in this projects. Afterwards the projects on Active Object Database Systems (AODBMS) and their research directions are reviewed.

This paper is organized as follows. In section 2, we discuss the rules in active database systems. In section 3, we give the application areas of active databases. The research issues lead by these advanced applications of ADBMS are presented in section 4. Section 5 gives the survey of reported work in the area of application modeling in ADBMS. The work on execution modeling and concurrency control of active database transactions has been reviewed in section 6. Important projects on active RDBMS and active ODBMS and the research directions pointed out by these projects are reviewed in section 7 and section 8 respectively.

2 Rules in Active Database System

A consensus seems to be emerging among the database community about the elemental components of a rule used for supporting active functionality. These components are an event expression, one or more conditions, an action, and a set of attributes. A rule with these components is called an

Event-Condition-Action rule or ECA rule [63]. The components of a rule are usually packaged in three different formats:

1. EC-A where only the action and the condition are specified (events are implicit)
2. E-CA where condition and action are specified together, event is specified explicitly and
3. E-C-A where all three components are explicitly specified.

In Sybase [3] and Interbase [2], a rule is composed of an event part and an action part (format 2). The action part is a transaction, i.e. a sequence of Transact-SQL statements in Sybase or a sequence of GDML statements in Interbase. The condition part is encoded as part of the action. ETM (Event Trigger Mechanism [146]) also uses the same format. On the other hand, Postgres [184], Starburst [201], and HIPAC [64] have separate event, condition and action parts (format 3). An extension of the above scheme includes a contingency action to replace the specified action when the action cannot be completed within a specified deadline [64]. Production rule systems (e.g. the expert system OPS5) typically use the first format.

For example, the following Starburst rule aborts the transaction whenever the average of updated employee salary exceeds 100:

define rule *AverageTooBig*
on update to Employee.salary *[⇐Event Part]*
if ((select avg(salary) **from new-updated)** > 100) *[⇐Condition Part]*
then rollback *[⇐Action Part]*

The rule *AverageTooBig* is specified in the third format, where separate event, condition and action can be specified.

2.1 Rules in Commercial RDBMS Systems and SQL3

Most commercial relational database management systems (RDBMS) support active database rules, referred to as *triggers*. These RDBMSs suffer from the following shortcomings:

1. These RDBMSs lack standardization in their syntax and semantics of trigger. Thus applications written using triggering facilities become unportable.
2. These product don't have a clearly defined execution semantics.
3. Lack of advanced active database features like event composition, binding of events to condition and condition to action, coupling mode of rules and parallelism in rule execution, application specific events etc.
4. Most of the time these products have a limitation on the number of triggers that may be defined, or on the interaction between the triggers.

In this subsection, we discuss in brief the active facility in the prevalent commercial relational database system like Oracle, Sybase, Ingres and Informix. To provide a more consistent support for active mechanisms in relational systems, assertions and triggers are included in the emerging SQL-3 standard [200]. Thus we start our discussion with SQL-3, to see what standard bodies are doing in this area.

2.1.1 Active features in SQL3

SQL-3 standard (which later became SQL99) support active functionalities by means of (a) assertions and (b)triggers [148].

Assertions in SQL-3: The SQL-3 assertion facilitates for checking specific constraints (on a table) during the transaction execution. The constraint can be arbitrary SQL predicate and the assertion is satisfied if the constraint evaluates to true. The constraint is evaluated when explicit *assertion events* occurs. The assertion events may be transactional events (e.g. "BEFORE COMMIT") or data modification events (e.g. insert/delete/update on a table). The granularity of constraint evaluation can be *tuple-level*, i.e., the condition is evaluated for each tuple of the specified table. Otherwise, granularity of evaluation is *statement-level*, i.e., the condition is evaluated exactly once for the entire table. A constraint's evaluation may be *immediate* if the constraint is evaluated after every SQL statement that may affect the constraint or it may be *deferred* if constraint checking is not performed until the commit point of a transaction.

Triggers in SQL-3: SQL-3 follows the active database ECA paradigm. The event is the monitored database operation, the condition is an arbitrary SQL predicate, and the action is a sequence of SQL procedure statements. These procedure statements are executed iff the event occurs and the condition evaluates to true. Like assertion in SQL-3 standard, trigger facility supports both row-level triggers (with a transition granularity of tuple) and statement-level triggers (with a transition granularity of set). Statement-level triggers are executed once in response to an update operation on a table, no matter how many tuples are affected by the update. In the condition and the action part of a trigger, references to values of the tuple/table before and after the execution of trigger are also available. However, perhaps the most important feature of the SQL-3 standard is that it makes explicit how triggers are to interact with other features found in relational databases, and in particular, declarative integrity-checking mechanisms. For example, SQL-3 prohibits triggers on tables with referential integrity constraints.

The syntax and execution behavior of commercial RDBMS systems span a broad spectrum in their variance from the proposed SQL-3 standard and also they widely

vary among themselves too. We discuss this in the following subsections.

2.1.2 Active features in Oracle

The RDBMS Oracle 7.0 and higher versions supports triggers [200]. The triggers in Oracle may be specified to be executed before or after the triggering operation, and with either tuple-level or statement-level granularity as in SQL3. The condition part can be specified only for tuple-level triggers and is restricted to be a simple predicate on the modified tuple. The triggered action is a procedure block written in PL/SQL, a special database programming language supported by Oracle. The old and new value of the tuples are available only in case of tuple-level triggers. Triggers may cascade as a result of recursive trigger invocation, but there is a maximum limit on the number of cascading triggers. The trigger processing may be non-deterministic as the system does not guarantee any particular order for the row processed by the SQL operations. In Oracle, triggers are used for auditing and event logging, automatically computing derived data, enforcing referential integrity constraints and maintenance of replicated table synchronously.

2.1.3 Active features in Sybase

In Sybase 10, triggers are only statement-level and can execute after the triggering operations [200]. Triggers are supported for insert, delete and update operation on a table. The old and new values of the tuple are available by two system defined temporary tables called INSERTED and DELETED. These table include all the tuple that were inserted or deleted by the triggering operation. Updates are treated as a delete followed by an insert. Triggers in Sybase can cascade with a built-in limit of 8 firings. Triggers in Sybase can be used for implementing referential integrity constraints like "cascaded delete".

2.1.4 Active features in Ingres

In Ingres, triggers, referred to as rules, are executed after its triggering operation and with a tuple level granularity [200]. The rule triggering events are insert, delete or update on a table. The condition part of a rule may reference current, old and new attribute of the tuple that cause the rule to fire and these values may be passed to the procedure in the rule's action. Like Oracle and Sybase, rules in Ingres may cascade upto 20 rules. If an error occurs, during rule execution then the triggering operation and all subsequent trigger actions are rolled back. Rule are used in Ingres for maintenance referential integrity constraints, enforcing of general-purpose business policies and for maintaining authorization schemes.

2.1.5 Active features in Informix

In Informix rules, multiple triggers may be defined within a single rule. Thus separate condition-action pairs may be defined together, one pair to be triggered before a statement, one pair to be triggered for each row affected by a statement, and one pair to be triggered after a statement [200]. The triggering events may be insert, delete on a table or an update on a column of a table. The old and new values of the tuple can be referenced in the rules condition and action part. Actions are arbitrary sequence of insert/delete/update statements or procedure calls. Like Oracle, triggers in Informix can cascade upto with a built-in limit of 60. When logging is enabled, if an error occurs during trigger execution then the triggering operation and all subsequent triggered actions are rolled back. When logging is not enabled, handling of errors in trigger execution is application's responsibility.

2.2 Rule Expressiveness

As we have discussed in section 2.1, commonly used data manipulation languages (SQL and its variants) currently support primitive ECA rules. As a result, *ad hoc* extensions have been proposed even for the specification of simple integrity constraints (e.g. domain constraints). Some of the earlier approaches specifically addressed enforcement of integrity constraints by extending the data model for specifying triggers.

In contrast, active DBMSs require enhancement to the data model in at least two ways:

1. for specifying events
2. for specifying conditions and actions along with the attributes for tailoring the behavior of rules.

2.2.1 Event Specification Enhancement

In ADBMS, in addition to the specification of events corresponding to database operations, such as insert, delete, and modify, specifications of composite events, temporal and/or periodic events as well as external events need to be supported [72]. This has to be provided by defining event types and event consumption policies.

Event Types: The event types may be *primitive* or *composite*. Primitive event types define the events which are raised by a single low level occurrence, for example, method invocation, data item modification, transaction operation, abstract and time event types. Composite event types are defined by some combination of primitive and composite events, using set of event constructors such as disjunction, conjunction, sequences etc. Rich event algebras have been proposed for a range of systems, including HiPAC [63], ODE [103], Sentinel [48] and SAMOS [94].

Event Consumption Policies: To detect composite event from a sequence of primitive events, the event consumption policies need to be specified. In [48] four possible event consumption policies have been defined namely *recent*, *chronicle*, *continuos* and *cumulative*. In *recent context*, most recent primitive events are used to construct a composite event. In *chronicle context*, primitive events are

consumed in a chronological order. Each primitive event starts the detection of all possible composite events in *continuos context*. In *cumulative context*, all the primitive events of same type are accumulated until the composite event is finally raised.

2.2.2 Condition and Action Specification Enhancement

Specifications of conditions and actions also require extension to the language supported by the data model. Conditions and actions may be complex, may include aggregation operators, and may refer to the old and new state corresponding to the state before the execution of the operations and the state created after the execution of the operations, respectively.

2.3 Rule Execution Semantics

The introduction of ECA rules, executed as a side effect of transaction execution, requires redefinition of transaction semantics. The semantic relationships among the original transaction and the subtransactions invoked for evaluating the condition and execution of action, are specified by *coupling modes*. Several alternative coupling modes have been proposed [124]. Suppose an operation O is performed by transaction T, and that the invocation of O is the triggering event E for a rule with condition C and action A. Assuming that the condition C is satisfied, it would be necessary to execute action A. There are three possible ways by which A can be executed:

- *Immediate:* The action A is executed immediately when the event E occurs and before the next operation in T. Thus, the processing of remaining steps of the transaction which caused the event to occur (i.e. T, the *triggering transaction*) is suspended until the evaluation of the fired rule has been completely processed.
- *Deferred:* The action A is executed after the last operation in T and before T commits. This type of coupling mode is required for enforcement of integrity constraints in a transaction oriented environment.
- *Detached:* The action A is executed in a separate transaction T'.

Furthermore, the action A does not have to be executed at the same point where the condition C is evaluated. For example, the condition can be evaluated immediately after the triggering event, while the action is executed in a separate transaction. The condition C and action A can be considered separately, subject to the constraint that A cannot be executed before C is evaluated. There are seven distinct coupling modes that satisfy these constraints:

1. Evaluate C and execute A immediately when the event E occurs and before the next operation in T.

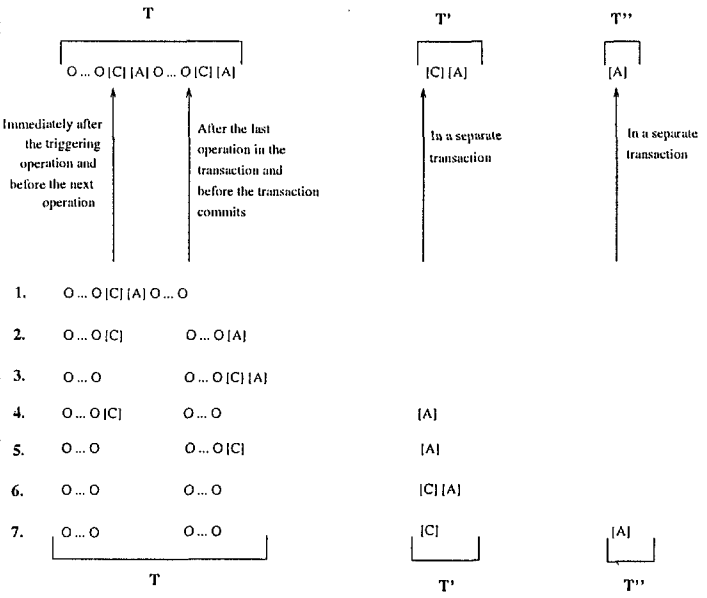


Figure 1: Different type of Rule Coupling in Active DBMS

2. Evaluate C immediately when the event E occurs and execute A after the last operation in T and before T commits.
3. Evaluate C and execute A after the last operation in T and before T commits.
4. Evaluate C immediately when the event E occurs and execute A in a separate transaction T'.
5. Evaluate C after the last operation in T but before T commits, and execute T in a separate transaction T'.
6. Evaluate C and execute A together in a separate transaction T'.
7. Evaluate C in one separate transaction T' and execute A in another separate transaction T''.

These cases are depicted in Figure 1. In addition to specifying the event, condition, and action, the definition of a rule must also specify the coupling mode of condition evaluation and action execution.

Rule Cycle Policy: The cycle policy of the rule execution model defines how the events signaled during condition and action evaluation need to be processed. In *iterative* cycling policy, the condition and action execution of the event signaling rule will continue even after the event signaling. In *recursive* cycle policy, event signaling during the condition and action evaluation causes the event signaling rule to be suspended, so that any triggered rule due to this event can be fired. Thus recursive cycling policy needs to be supported for immediate mode rule execution, and iterative cycling policy for deferred mode rule execution.

Rule Scheduling: It is possible that an event occurrence can make several rules eligible for execution. The scheduling of rules determines (a)the selection of the next rule(s)

to be fired and (b) the number of rules to be fired. Next rule selection policy is often carried out by means of priority mechanism in which rule priorities are specified using one of the following ways:

- using a *numerical* value for each rule [188], which is its priority.
- specifying *relative* priorities of rules by stating explicitly that a given rule needs to be fired before another when both are triggered at the same time [7].

The number of rules to be fired from the triggered rules can have following options:

(a) To fire all rules sequentially. This policy is needed for rules supporting integrity maintenance.

(b) To fire all rules in parallel, to facilitate more efficient rule processing as in HiPAC [63].

(c) To fire all instantiations of a specific rule before any other rules are considered, which is mostly used for expert systems to yield more focussed inference mechanisms.

(d) To fire only some rule instantiation(s). This policy is used to support derived data in which out of several derivation criteria only one need to be selected.

2.3.1 Dependencies of Rule Expressiveness and Rule Execution Semantics

Rule expressiveness and rule execution semantics are dependent on each other. In [163], a formal framework was introduced for studying the semantics and expressiveness of active databases. This framework allows to provide insight into the interplay of various active database features, and their impact on expressiveness and complexity. It was found that:

- Unbounded immediate triggering has a complexity of EXPTIME. But if the depth of nesting of immediate triggering is bounded, then the complexity goes down to PSPACE.
- Deferred triggering is computationally more powerful than immediate triggering. Several complexity bound may be found in case of deferred triggering under various restrictions on queuing discipline. If multiple occurrences of rules are not allowed in the triggered rule set queue, then the complexity becomes PSPACE. If multiple occurrences of the same rule is allowed in the queue, but not with same delta relations as parameters, then the complexity is EXPSPACE.
- For mixed triggering, i.e., when immediate and deferred triggering are combined, it subsumes both.
- Also, the complexity results induce relative expressive power of various restrictions. Thus, relative expressive power of different active databases can be determined based on their coupling mode supports and rule queue management strategy, as studied in [163].

3 Applications of Active DBMS

Since an ADBMS is intended to implement a greater portion of application semantics within the DBMS, it is necessary to investigate possible application areas. Below we look into some of the applications that have been implemented on top of an ADBMS.

- **Constraint and Integrity Management :** Constraint management and integrity control are classical application areas for ADBMSs. Constraints on the database are expressed as first-order predicates. Using these constraints, a set of production rules are generated (possibly redundant and non-terminating) which are able to take corrective actions corresponding to violated constraints. In [37], Ceri et. al. have introduced a scheme for automatic correction of violated constraints. A rule analyzer selects a subset of generalized rules such that termination is ensured and a large number of constraints are compensated. Static analysis algorithms described in [9, 8] determine whether a set of rules applied to a given user transaction will either terminate producing a unique final database state (*confluence*), or produce unique and user observable side effects (*observable determinism*). These algorithms have been based on the Starburst rule system [201, 202] and can be used by other rule evaluation systems that allow arbitrary execution points within a user transaction. Decidability and undecidability results for the termination problem of active database rules has also been investigated in [13]. The PARDES system [81, 82, 83] uses an integrity constraint approach based on an object-oriented ADBMS. The rule semantic is 'ON update AND condition DO update'. In PARDES, the rules are triggered not by events but by modifications of data. It addresses a set of semantic problems such as (i) non-determinism caused by different rules concerning the same object, (ii) infinite loops, and (iii) direct update problem, where both rule execution and regular updates may modify an object. In [42] production rules of an ADBMS have been used for defining a rule set for view maintenance. Here the user defines a view as an SQL select expression. The system automatically generates the production rules to maintain the materialization of the views. In [107], production rules have been used for integrity repair and for detection of integrity constraint violation. The rule language introduced in PARDES [107] is powerful enough to specify violation of repair actions. The Postgres system [185, 186, 187, 188] uses rules for maintenance of views, integrity control,

and protection. Usage of production rules for maintenance of temporal conditions and integrity maintenance have been proposed in [180], where incremental detection algorithms for complex temporal conditions have been given. Here, the temporal conditions are monitored as the database state changes with time. Usage of rules for database internal applications has also been discussed in [76].

- **Workflow Management :** In [35], ECA rules are used to specify the coordination of tasks within a multi-step workflow. Coordination aspects of tasks are synchronization of steps and mechanism to guarantee the right number of step execution within the workflow. In [65, 66], it has been shown that complex transaction models for workflow execution can be defined using rules. HiPAC transaction coupling modes have been extended by special transaction dependencies which are useful for dependent and nested top transactions. An example workflow model for a hospital patient information system has been defined using these rules [65].

- **Cooperative Problem Solving :** In a cooperative problem solving system, events and context information of long running activities have to be persistent [52]. Complex cooperation dependencies are captured by algebraic expressions involving those events and contexts. It has been shown that an active database may be used as a persistent blackboard for basic and complex event monitoring. The active database performs the tasks of event collection, monitoring complex events, and notification to the application layers.

- **Cooperative Work :** Distributed cooperative task execution can be done using ECA rules of an ADBMS [105, 145]. In distributed cooperative execution, the cooperation knowledge for execution of the task is coded in the individual agents. Using ADBMS, the policy of each of the agents is coded using ECA rules. The underlying active DBMS thus controls the cooperative task execution. ADBMS functionality may also be used for coordination among agents [19].

- **Multidatabase System :** Active DBMS production rules may be used to specify data integrity constraints in heterogeneous distributed multidatabases having different local and global data integrity constraints [73]. It is based on an architecture that wraps each local DBMS and a framework allows different levels of cooperation.

- **Advanced Transaction Model :** Flexible transaction models can be developed using an active DBMS as the building block [51, 104]. This approach presents an application for active ECA rules to specify the behavior of the intended DBMS. In [51], a framework has been developed where ECA rules are used to implement different transaction models within a DBMS. The different locking data structures for concurrency control are updated by the ECA rules according to the specified transaction model. The ACTA framework [56] has been used for specification and reasoning about transaction structure and behavior in advanced transaction models. In ACTA, a set of logical rules are used to specify the advanced transaction models. ACTA has been used in [204] to reason about transactions and data timing constraints in real-time active databases for analysis and verification of real time software. A rule based extensible DBMS kernel has been developed in [104], to support different transaction manager semantics described by a set of logical ACTA rules, which are translated into appropriate ECA production rules.

- **Finance :** The requirements to be met for implementing financial applications using an Active DBMS have been discussed in [54]. Applications of commercial trigger database system (SYBASE, INGRES, ORACLE) to business rules in information systems have been discussed in [140] with the case study in the area of banking and insurance. It has been shown that ADBMS can be used to implement business rules of financial applications.

- **Computer Integrated Manufacturing (CIM) :** In CIM, execution and coordination of multiple activities are required. The execution of the activities are based on outcome of previous activities and also the different triggering conditions based on the environmental parameters. Usage of temporal active databases for CIM applications have been discussed in [74]. Coordinating CIM activities in a multidatabase system environment has been taken care of in [84].

- **Graphical Interface :** Dynamic displays can be supported using active rules [70]. Dynamic displays having graphical interface react automatically to modifications of the underlying database and can be implemented without changes to the application system or the graphical interface.

4 Research Issues in ADBMS

The advanced applications discussed in section 3 requires the effective support of ECA rules in an active DBMS, which leads to the close analysis of the following major issues [62]:

- **Knowledge Model:** extend conventional data models to accommodate ECA rules and their associated execution and timing requirements as far as possible. This issue has been investigated by several researchers [62, 64, 188, 11, 126, 50, 32, 31].

- **Application Model:** develop an application modeling framework for applications running on top of an active DBMS. Obviously, such a framework will differ

from existing modeling frameworks for applications running on a passive DBMS since here, the database is responsive and can trigger the application transactions which they have opted for. Though considerable effort has been directed towards knowledge modeling, not much work has been reported on application model development.

- **Execution Model:** extend conventional transaction models to specify correct interleavings of the system-triggered actions in addition to user or application-initiated transactions. Transactions subscribe to the database for the events they are *interested in*. Transactions containing the rules, would spawn and signal sub-transactions as a result of triggering of the rules. Streams of long running inter-communicating and co-operating hierarchically structured transactions can be constructed in this manner. This suggests the need for a new framework for handling communication and cooperation among transactions. Though some work has been carried out on execution modeling of active DBMS transactions [124], issues related to communicating and cooperating active DBMS transactions in the presence of detached mode rules however, need to be carefully examined.
- **Scheduling:** develop algorithms for scheduling tasks to satisfy concurrency, cooperation semantics, and timing constraints. This has to be done based on appropriate execution model for cooperating and communicating transactions.
- **Condition Monitoring:** develop techniques for efficiently evaluating sets of dynamic, overlapping conditions.
- **Architecture:** define the functional components of an active DBMS, and their interactions.
- **Performance Evaluation:** construct a test-bed for evaluating alternative architectures and algorithms for condition monitoring and scheduling.

5 Application Modeling on Passive and Active DBMSs

Research in ADBMS has concentrated mostly on event modeling [53, 94, 95, 103], transaction modeling [62, 65, 49, 31], and execution modeling [124, 66, 31]. But in order to develop complex applications, we need to consider an application modeling framework where all these aspects are simultaneously taken care of. Brodie et. al. [26, 27, 139, 157] have proposed several techniques for dealing with database-intensive applications on relational databases. In Brodie et. al [26, 27], transactions operate on a classical relational database, i.e., the RDBMS is assumed to be not an active one. Recent research on dynamic modeling of data and systems has resulted in several representation formalisms [10, 139, 158]. In the DATAID project [10], programmer-defined forms are used for event specification and these are converted to Petri-Nets for analysis. ACM/PCM [158] uses a behavioral specification language based on predicate transformers. The event model of King and McLeod [139] uses design schema or scripts. In the TAXIS project [158], the TAXIS language, which is similar to an object oriented programming language, is used to specify transactions. In all of these schemes, the database is considered as passive and transactions operate on the database.

5.1 Need for Object Oriented Modeling

Non-standard applications have to cope with frequently changing requirements which are, to a large extent, due to changes in the business environment [152]. Those aspects of the business environment which are subject to frequent changes are often referred to as *business policies*. Business policies typically capture context-dependent and time-dependent organizational knowledge. They may be based on ethics, law, culture and organizational commitments by either prescribing a certain action or by constraining the set of possible actions [122, 159]. To cope with changing business policies, it should be possible to easily adapt the application implementing the respective policies. The introduction of object oriented paradigm has been one step in this direction. Object-oriented languages and development environments help to intuitively model the universe of discourse and to adapt to the changing requirements [87]. However, a mechanism for explicitly specifying business policies in a natural and straightforward way is still missing. In most object-oriented systems, business policies are implemented by some methods, or part thereof, and thus business knowledge is mixed with code realizing basic functionality with a low modification probability. In this respect, the authors in [171] have suggested distinguishing methods containing policy ("making of context dependent decisions"), from methods containing implementation ("execution of fully specified algorithms"). It has been shown, however, that a more intuitive way to describe business policies is in terms of rules, since the domain specialists usually do so in expressing their system requirements [162].

5.2 Integrating Rules with Object Oriented Modeling

Approaches for integrating rules with object-oriented concepts have been widely accepted as a powerful means for explicitly modeling applications [159, 170, 191]. These approaches are also known as rule-based models, or more accurately as *active object oriented models*. In these models, ECA rules are often used for explicit, localized, and transparent specification of business policies of applications [71, 72]. The usage of ECA rules enhances the reusability of methods and classes since such methods and classes

need not contain application specific policies. Moreover, since a business policy can be expressed by rules, it can be readily modified if the situation demands so.

Active Object Oriented Database Systems (AODBMS) provide both object-oriented and rule based specification of the applications. Thus in an AODBMS application, the database is an integral part of the dynamic model of the application. Therefore, the methodology for traditional transaction modeling schemes has to be modified to capture the dynamic properties of the application. In [134, 131, 132], a methodology has been reported for modeling application Active Object DBMS (ADBMS). The proposed methodology in [134] investigates the interrelationships among transactions, objects, and events in a closed loop environment. A database is considered not only as a repository of the application data, but also as a storekeeper and monitor of the control and dynamic aspects of the complete system. Thus a database itself may automatically generate transactions as a part of the application semantics.

The conceptual design of AODBMS applications can be carried out using multi-level diagrams as proposed in [177]. When a rule set of an ADBMS becomes quite large, the rule set needs to be modularized for better conceptualization and designing. Several modularization techniques for active rules design have been investigated in [14].

6 Execution Modeling in Active DBMS

In this section, we review the reported research on execution modeling of active DBMS transactions. The execution model of ADBMS transactions also determines the concurrency control scheme to be used for concurrent execution of multiple transactions. The research on concurrency control of ADBMS transactions has naturally been influenced by classical concurrency control protocols for atomic and extended transaction models.

6.1 Atomic Transactions :

Traditional transactions [77, 109] are based on the notion of *atomicity* and thus are often referred as *atomic transactions*. Atomic transactions enforce the properties of *(i) failure atomicity, (ii) consistency, (iii) isolation, and (iv) durability*, commonly referred to as ACID properties. *Failure atomicity* means that either all or none of the transaction's operations are performed. *Consistency* means that a transaction maintains the integrity constraints of the database. *Isolation*, commonly known as *serializability*, means that concurrent transactions execute without any interference as though they were executed in some serial order. *Durability* means that all the changes made by the committed transactions become permanent in the database.

In most of the schemes, serializability is based on the notion of conflicting operations and is called *conflict preserving serializability*. Two operations conflict when their effect is order-dependent. Conflict preserving serializability ensures that pairs of conflicting operations appear in the same order in the equivalent serial schedules [77, 110]. Different versions of serializability such as *view* and *state* serializability, have been introduced in [161]. Although view or state serializability may increase concurrency as compared to conflict preserving serializability, it is NP-complete to test whether a schedule is view or state serializable. Both pessimistic and optimistic schemes have been proposed to ensure serializability in centralized and distributed environments. Most of these schemes are based on either two-phase locking or timestamp ordering [20]. To achieve failure atomicity [21], two schemes are commonly adopted. One is based on the notion of careful replacement (commonly referred to as intentions lists) [193] and the other is based on maintaining execution logs [112]. The intention list based recovery applies the proposed updates of a transaction when the transaction decides to commit. If the transaction is aborted, then it need not be undone since the proposed updates have not been applied to the database yet. Also, if the transaction is committed, but the system failure occurs before posting its updates, then the intention list of the transaction is applied on the database, i.e. the transaction is redone. Thus it is also called "No Undo but Redo" based recovery [21]. The execution log based recovery applies the effects of the transaction to the database as the transaction executes the operations. It also keeps log of the "old" values of the objects. If the transaction decides to abort or a system crash occurs, then all the old values are brought back for the unfinished transactions and all the updated values are put back into the database for the committed transactions. This approach is also called "Undo and Redo" based approach [21].

6.2 Extended Transaction Models

In atomic transactions, interleaving of transactions is *implicitly* constrained by the concurrency properties of the objects, i.e. operation conflicts. Here, the consistency of the database is based on the notion of serializability. In an alternative approach, interleaving of transactions is *explicitly* constrained by the concurrency specifications of the transactions. This approach may not achieve serializability, but can be used in such a way that consistency can be preserved. In such schemes, concurrency specifications are defined according to the semantics of the transactions and the data they manipulate.

In recent years, this approach has resulted in a proliferation of extended transaction models [75, 4], since this approach provides the only means for dealing with the functionality and performance requirements of the new advanced applications. Here we provide a brief overview of some well-known extended transaction models.

One of the most well-known extended transaction models is the Nested Transaction Model [156, 164] in which a transaction is composed of subtransactions that may execute concurrently. A subtransaction can be further de-

composed into other subtransactions, and thus, a transaction may expand in a hierarchical manner. Subtransactions are executed atomically with respect to their siblings and other non-related transactions and are failure atomic with respect to their parent. The subtransactions can abort independently without causing the abort of the whole transaction. Thus nested transactions are designed to localize failure within the transactions and can exploit parallelism within transactions. A subtransaction can potentially access any object that is currently accessed by one of its ancestor transactions. In addition, any object in the database is also potentially accessible to the subtransaction. When a subtransaction commits, the objects modified by it are made accessible to its parent transaction. However, the effects of objects are made permanent in a database only when the root transaction commits.

Another scheme of extended transaction model has been proposed in [89]. It is based on the notion of compatible transactions. In this scheme, pairs of transactions are distinguished as being *compatible* or not. Compatible transactions, as with compatible operations, have semantic structures that allow them to execute concurrently. This scheme simplifies the specification of compatible transactions by classifying them into different semantic types.

The concept of compatible transactions has been generalized to several levels leading to the notion of *multi-level atomicity* [153]. In this model, transactions can belong to more than one semantic type. Each transaction type has different sets of breakpoints, inserted between the steps of a transactions at appropriate points. Steps of compatible transactions can be interleaved at these breakpoints. In [153], the breakpoints are embedded in the body of the transactions. In [181], breakpoints are external to the transactions, captured by a set of *patterns*. A pattern is a *state transition diagram* [16] which expresses the goals and pieces of work to be done. In this approach, *cooperative transactions* are grouped into *transaction groups*. A transaction group represents the unit of consistency and recovery. Components of cooperative transactions may not produce consistent results. In fact, cooperative transactions do not have any properties of atomic transactions and just represent different threads of controls. The consistency constraints for transaction groups are specified with the help of the "patterns". Since goals in a pattern may invalidate the goals of other patterns, *conflict specifications* among the operations are needed to control the interleaving of concurrent patterns. Thus the effects of a transaction group are considered to be consistent as long as all the steps of applicable patterns are executed (invoked exactly once by some cooperative transactions in the transaction group) and the patterns are interleaved concurrently.

Just like transaction groups, *split transactions* [165] and *sagas* [88] have also been proposed to deal with the problem of long lived and cooperative transactions. In neither of these schemes the notion of compatibility between transactions has been explicitly used. A saga is like a two level nested transaction with traditional transactions as child transactions. Each child transaction is associated with an application specific compensating transaction. A saga can be interleaved in any way with other sagas, but it cannot be partially executed. If a saga is interrupted, it attempts to proceed by executing the contingency transactions (*forward recovery*), or amends partial executions by invoking compensating transactions (*backward recovery*). A contingency transaction [29] is invoked upon the failure of a transaction to accomplish a goal similar to that of the failed transaction. Formal aspects of compensating transactions are discussed in [142, 149]. A variation of sagas [90] allows the characterization of children as *vital* or *non-vital* where the abortion of a *vital* child causes the abortion of the transaction. Sagas are appropriate in applications where each child transaction does not have to observe the same consistent database state.

In the context of long durational and cooperative transactions, the *Recoverable Communicating Actions (RCA)* model has been proposed to deal with the problem of non-hierarchical computations [194]. In this model, an action, the *sender*, is allowed to communicate with another action, the *receiver*, by exchanging objects resulting in an *abort-dependency* of the receiver on the sender. If the sender aborts, then the receiver must abort as a result of dependency. However, partial failures are tolerated since an action may abort without aborting the action on which it has developed an abort-dependency.

The concepts *compound transactions* [176] and *cooperative transactions* [144, 143] use the same correctness criterion, which permits non-serializable executions while satisfying the individual postconditions of the transactions. In the case of the compound transactions, the criterion are known as setwise serializability, whereas in the cooperative transactions case it is known as *predicatewise serializability*. Objects in both schemes are grouped into sets based on the database consistency constraints. In the compound transaction scheme, it is assumed that each set, called an *atomic data set*, has independent consistency constraints and that individual transactions operate on a particular atomic data set. There are no such assumptions in the case of cooperative transactions. Thus it is possible for cooperative transactions to be serialized in different orders with regards to different atomic data sets.

Finally, to facilitate the formal description of transaction properties in an extended transaction model, a comprehensive transaction framework named ACTA [57, 55] has been developed. Using ACTA, one can specify and reason about the nature of interaction among transactions in a particular model. ACTA characterizes the semantics of interactions (i) in terms of different types of dependencies between transactions (e.g., commit dependency and abort dependency), and (ii) in terms of transactions' effects on objects (their state and concurrency status i.e., synchronization state). Through the former, one can specify relationships between significant events, such as *begin*, *commit*, *abort*, *delegate*, *split* and *join*, pertaining to different transactions. Also, conditions under which such events can

occur can be specified precisely. Transactions' effects on object state and status are specified by associating *view* and *conflict set* with each transaction and by stating how these are affected when significant events occur. A view of a transaction specifies the state of objects visible to that transactions. The transaction's conflict set contains those operations with respect to which conflicts need to be considered. Thus the ACTA framework can characterize transaction models that can associate different semantics with the notions of *visibility*, *consistency*, *recovery*, and *permanence*. Also, ACTA can reason about the properties of the transaction models.

6.3 Transaction Model and Concurrency Control of Active DBMS Transactions

There have been several research efforts on rules, rule control, and transaction framework in ADBMS [124, 186, 63, 115, 173, 202, 17, 178, 43, 166]. The concurrency control schemes adopted in these projects are determined by the transaction models of the ADBMS transactions in presence of triggered rules. The rule subsystems of active DBMSs, POSTGRESS [186], Ariel [115], HiPAC [63], and Alert [7], support *priorities* to define control structures among rules. However, priority based rule control is not flexible enough to support the control requirements of rules in advanced DB applications. Also, these rules systems do not support cooperative transactions, i.e. where the partial effect of a transaction may be visible to some other concurrently running cooperating transactions.

In [178], RDL1 supports a set of control constructs namely 'sequence', 'disjunction', and 'saturate', for specifying the control structures among rules. In RDL1, rules are defined in modules. Each module contains a rule section in which rules are defined, and a rule control section in which the control structure among these rules is defined. However, a set of rules defined in a module cannot follow different control structures when they are are triggered by different events. For a set of rules to follow different control structures when they are triggered by N different events in RDL1, the same rule must be defined N times in N different modules.

In an active DBMS environment, the rule execution must be uniformly incorporated into a transaction framework [172]. In POSTGRES and Ariel, the execution of rules has been incorporated into a flat transaction model. In [124], Hsu et. al. have described a more expressive model which is basically an extended nested transaction model to capture rules and nested triggering of rules. A subset of this execution model was analyzed in [36] to study the system performance as a function of transaction boundary semantics for various level of data contention, rule complexity, and data sharing between externally submitted tasks and rule management tasks. These results demonstrate that the way in which transaction boundaries are imposed can have a major impact on the performance of an active DBMS, and that this aspect of rule semantics must therefore be carefully considered at the time rules are specified. Other significant works that use variants of the nested transaction model for modeling the execution of rules are reported in [17, 43, 166]. However, the tree based structure of nested transaction models is not expressive enough to capture complex control dependencies among rules in a uniform fashion. In [130], a graph-based transaction model has been defined which captures triggered rule graphs uniformly and places them at an appropriate control point of the transaction structure according to the occurrence of triggering actions. Hence, a nested transaction model based execution becomes a special case of this graph-based model. Alternative transaction models reported in [58, 195, 30, 12], however do not focus on the uniform incorporation of rules, rule control, and trigger occurrence in the transaction framework. Also, all the previously mentioned works on execution model do not support cooperation among the transactions since they are essentially closed transaction models (nested/flat) in general.

Development of knowledge models has received considerable attention especially for increasing the expressive power of event specification languages without consequences for the execution models. In these works, one of the most challenging problem is the development of a transaction model involving composite events, i.e. events representing a combination of other events [31, 48, 145]. The nested transaction model has been suggested as an appropriate transaction model for AODBMS involving both primitive as well as composite events[137].

6.3.1 Nested Transaction Models for Primitive Events

Most of the execution models of existing AODBMSs are based on nested transaction models [31, 48, 59, 91]. However, these nested transaction models supports closed nested transactions [5]. The nested transaction model is particularly appropriate for the active system due to the following reasons [124, 160]:

- A nested structure accommodates nicely the hierarchical composition of execution units as it is in the case of active systems. The relationship(s) between *event signaling transaction* (the transaction in which an event is raised) and one or more *rule transactions* (the transactions wherein the condition and/or action are processed) can naturally be mapped to the relationship(s) between *parent transactions* and *child transactions* within a nested transaction model.

- Nested transactions preserve serializability and top-level atomicity while allowing for a decomposition of a "unit of work" into subtasks as a prerequisite for intratransaction parallelism. This is especially useful for active systems since rules often realize add-on functionality which may be executed parallel to the event signaling transaction as well as to each other [179].

- Lastly, as compared to traditional "flat" transaction models, nested models offer finer-grained control over

concurrency and recovery [121], thereby allowing rollback of rule transactions to be executed independently of event signaling transactions.

The different variations of nested transaction models used in active systems work well with primitive events. A primitive event triggering a rule establishes a one-to-one relationship between the event signaling transaction and the rule processing transaction corresponding to the condition and/or action of the rule. The transaction coupling for such one-to-one relationship is defined by means of coupling modes [31, 65]. A coupling mode, in most systems, permits two types of coupling among the event signaling and rule processing transactions. The rule processing transaction may be executed serially either in a subtransaction of the event signaling transaction or as an independent top level transaction.

As discussed above, the event processing in these schemes is intra-transactional in nature, i.e. the events generated by a transaction T are processed in the context of T only. Any external transaction (say T') cannot make the events generated by T, an 'event of interest' for it. Therefore, a transaction cannot specify its 'events of interests' which are generated by other transactions and define the corresponding rules to be executed on occurrence of these events. When this kind of specification is allowed, then the event processing becomes inter-transactional, i.e. a transaction generated event may cross application and transaction boundaries.

6.3.2 Nested Transaction Models for Composite Events

Many active DBMSs allow composite events to be constructed from primitive events and/or other composite events using different kinds of logical operators such as conjunction (AND), disjunction (OR), and sequence (;) [48, 94]. A composite event consists of *component events*. The event starting the detection of a composite event is called *initializing event* and the event terminating the detection called the *terminating event* or *triggering event*. The transaction(s) wherein these component events occur are analogously called *initializing transaction* and *triggering transaction* respectively. Following this terminology, both an initializing transaction and a triggering transaction are special kinds of event signaling transactions.

With composite events, more than one component events may have to occur to trigger a single rule. Depending on the origin of these component events, different cases can be identified: all the component events occur in a single transaction, or some of them occur in different transactions. In the latter case, there are two possibilities, either the event signaling transactions are sequentially processed within a single thread of control, or they are executed in different threads of control. Note that in this case the composite event is *inter-transactional* since the component events would be received from different transactions running in separate threads. These transactions may belong to different applications running over the same database. Any of these cases would result in a many-to-one relationship between event signaling transactions and a rule transaction. This many-to-one relationship is specified by associating a transaction mode or coupling mode with the rules. The transaction mode specifies the semantic relationships among triggering events, condition, and action.

In existing active systems, the many-to-one relationship is handled in several different ways. There are some systems which do not allow component events of a specific composite event to occur in different transactions [48], i.e. they offer only the detection of intra-transactional events. Again, there are systems which allow component events to occur in different transactions. However, in some of these systems the transaction mode refers only to the triggering transactions [94]. Another class of systems allows transaction modes that refer to all event signaling transactions to be specified, but do not allow a subtransaction mode to be specified [31]. Finally, there are some systems which allow transaction mode for each event part of the composite event to be specified within the context of a rule [137, 136].

In the literature, only some systems deal with different transaction modes for composite events. Among these are REACH [31], SAMOS [92], Sentinel [48], Ode [102, 150], and TriGS [137, 136, 138]. In the following, the types of transactions supported by these systems are examined. All the five systems mentioned above allow component events, which are part of a certain *composite event*, to be signaled within *multiple transactions*. In SAMOS and Ode, composite events internally are represented within a complex database object structure. Thus, composite events on the one hand are signaled not before commit of signaling transactions. In SAMOS, they might even produce deadlock between event signaling transactions and, thus, prohibit its detection. In Sentinel, component events are flushed at commit of the corresponding event signaling transactions. Consequently, composite events represented in main memory are detected not before commit of all participating event signaling transactions. In contrast, REACH allows composite events to be signaled before commit of the participating event signaling transactions. Although Sentinel, TriGS, and REACH permit parallel composite event detection, Ode and SAMOS do not support this feature.

SAMOS uses a closed nested transaction model. However, due to restriction of the underlying object-oriented database system ObjectStore, neither subtransaction parallelism nor parent/child parallelism is provided for. Furthermore, SAMOS allows all types of coupling modes for nested rules. Sentinel supports nested transactions [31], where as REACH and Ode do not offer this facility.

One interesting feature of REACH is that it restricts the *combination of transaction modes* to some reasonable combinations to have better performance. Thus, rules containing composite events whose component events occur in a single transaction cannot be defined as immediate, since every time a component event is signaled it would have to be determined if it is a triggering event for an immediate

rule. Such rule can only be defined as deferred or detached. Similarly, rules containing composite events whose component events occur in multiple transactions are not allowed to be defined as immediate or deferred since it would be ambiguous which transaction is referred to. In REACH, various detached coupling modes are allowed for both, single and multiple event signaling transactions, provided that commit/abort dependencies are respected for all event signaling transactions.

In TriGS [137, 136], it has been shown that in the presence of composite events there is a need to apply transaction modes, not only to triggering transactions, but also to some or all of the event signaling transactions. TriGS has taken an approach which allows transaction modes to be specified for each of the component events in the context of a single rule. It has been shown that existing transaction models are not powerful enough to support a subtransaction mode with respect to multiple event signaling transactions. With this prerequisite, TriGS has extended nested transaction model by introducing multi-parent subtransactions. The multi-parent subtransaction model enables the semantic of composite events to be mapped onto appropriate transaction modes, i.e. parallel or sequential and dependent or independent execution. The transactions signaling events which are part of a composite event are allowed to cooperate in starting a common subtransaction in which the rule, i.e. its condition and/or action get executed. In TriGS's multi-parent transaction scheme, every parent transaction becomes commit-dependent on all of its children, i.e. the parent can commit only if all of its children transactions have committed. A child transaction becomes abort dependent on all of its parent transactions, i.e. if a parent aborts then its children have to be aborted. TriGS has enhanced the nested transaction locking protocol to ensure controlled upward and downward inheritance of locks [137].

The problem with the multi-parent transaction scheme of TriGS is that in a parent transaction, if a data value which has been modified by its committed child transaction is used, then the committed data may have to be rolled back because the child transaction is later aborted due to the abort dependency to another parent transaction. Upon abort, all such modifications would have to be rolled back within the parent. In case that such a parent transaction also had already committed, cascading aborts have to be performed. In order to keep track of which modifications are based on which child transaction, it was suggested that vital subtransactions [58] should be used within the whole transaction graph. A vital subtransaction identifies the parent to be abort-dependent on the child. This means that abort of a single vital transaction $T1$ would result in aborts of all transactions which are direct or indirect subtransactions of $T1$, or due to the abort semantics, direct or indirect parents of these subtransactions. The only transactions that would not be affected by these aborts are direct and indirect parents of $T1$. It was suggested in [137] that a reasonable solution to this problem has to be found.

The preceding discussions reveal that none of the AODBMS mentioned support horizontal cooperation, i.e. where a transactions effect may be read by other parallely running transaction based on some user defined transaction cooperation semantics. Vertical cooperation, i.e. parent-child cooperation, is supported by some execution models based on the extensions of nested transaction model. It was pointed out in [72, 179] that using vertical cooperation, cooperative applications like workflows, cooperative editing, etc. can be handled.

It was pointed out in [136] that how far ECA rules should be allowed to influence the application program needs to be carefully analyzed. On the one hand, database system should not be seen as a mere slave to the application, but rather should be able to autonomously control the application activities in various ways [72, 179]. On the other hand, it would also be possible that the application need not be aware of the underlying active capabilities if it chooses to do so. In [134, 133] these aspects have been discussed in detail. The proposed application modeling in [134, 131] gives a framework for designing the ADBMS so that applications can choose how much to affect the application activities due to the ADBMS. A cooperative nested transaction model which supports both controlled cooperation among the transactions and also provides facility to define nested and parallel control structures among the rules, has been discussed in [133, 135]. It supports vital transactions defined by the user to alleviate the problems with the existing execution models.

7 Active RDBMS Projects and Current Research Directions

7.1 Starburst

At IBM, the Starburst project attempted to extend a relational database system [199, 111] with an ECA rule system. The application of this research is introduced in IBM's DB2 database. Starburst rules were the first active rules with a clearly defined set-oriented semantics. Events, conditions, and actions are all expressed through an SQL-like syntax. An event is any update, delete, or insert command performed on some specified relation (or relational column). A condition is a predicate about the state of the database written in an extension of the SQL query language. A condition may also refer to the set of tuples affected by the event. For instance, if the event is a deletion of tuples from table T, the condition may refer to the set of tuples deleted by that action. The "transaction-affected" tuples are stored in *transition tables* which may be queried during evaluation of the rule condition. The condition may also be omitted, in which case the predicate is simply *TRUE*. An action is any non-empty sequence of SQL commands.

The set-oriented execution semantics of Starburst rules is presented in [201]. Starburst uses the deferred modes for

E-C couplings, and the immediate mode for C-A couplings. The Starburst rule language is based on arbitrary database state transitions of set-oriented execution semantics rather than tuple- or statement-level changes, yielding a clear and flexible execution semantics as discussed in [199]. But set-oriented semantics prohibits natural before option for rule triggering [199]. Also user may feel that they can better express rules using tuple- or statement-level granularity.

The implementation of Starburst rules is described in [202]. Starburst provides a mechanism, *attachment procedures*, which may be invoked after every access to a relation. Rules are triggered by designating an attachment procedure for every event relation combination specified in the event clause of a rule. When the specified event occurs, the associated attachment procedure is called and passes exactly the information requested by the rules to the transaction that caused the event. When the transaction is ready to commit, rule processing begins. Transition tables are formed on demand from the information reported by attachment procedures. Rule processing continues until no rule triggered by the transaction remains to be evaluated. At this point, the transaction commits.

Starburst work has also focused on the issue of rule selection and rule priorities. Since many rules can be triggered by a single event, some conflict resolution strategy is required to determine the order in which the pending rules are selected for execution. It has been shown that execution order of active rules affects the outcome in unpredictable ways [203]. To solve this problem, rule *priorities* are used so that the rule with the highest assigned priority is selected first for execution. The Starburst database system assigns a default priority (such as the time-stamp of rule definition) automatically. Users are able to override the default priorities with user defined priorities. The data structures and algorithms used to test and incrementally maintain user-defined rule priorities are described in [7]. Since predicting and understanding the behavior of active database rules are important aspects of any application development, a suitable methodology has been developed for statically analyzing sets of Starburst rules. These analysis determine (conservatively) whether a set of rules is guaranteed to terminate, and whether the rules are guaranteed to produce a unique final state [9].

The Starburst rule system has been used as a platform for developing a number of applications and for investigating various issues in active database systems, e.g. for integrity constraint maintenance [41], for maintaining materialized views [42], for implementing deductive databases [45], as well as for several other more ad-hoc applications. In [43], it has been shown how the Starburst rule system can be supported in a tightly coupled distributed database environment with full distribution transparency. Starburst also has been used for managing semantic heterogeneity in loosely coupled databases [44]. Some researchers have used the Starburst Rule System as a basis for studying and implementing secure active databases [182], dynamic integrity constraints [106, 190], and automatically generated compensating actions for static constraints [38]. Starburst work has also motivated the need for a formal foundation of active database system [198]. A formal foundation of active database rule languages would provide a very important step in understanding and characterizing the commonalities and differences across systems.

7.2 Postgres

The Postgres system can be considered as an object oriented relational system which extends the relational model with abstract data types, user defined operators and procedures, relational attributes of type procedure, and inheritance [169]. The first generation Postgres rule system, PRS-I [185], was subsequently modified to PRS-II, which can take care of old and new values of data items to deal with transition constraints [186, 187]. PRS-II rules have been used to solve the view update problem [188]. Procedural data types can also be simulated by the PRS-II rule system. It has been shown that by caching the action part, it is possible to improve performance and the scheme can be applied to materialize views [188, 189]. The main contribution of the Postgres work is that it shows that rules can be used to subsume views and procedures as special cases. This has lead to further research in these areas.

7.3 Ariel

Ariel is implemented on top of the EXODUS storage manager at University of Florida. Ariel focuses on extensions to the relational DBMS towards production rules with a forward-chaining rule system [120]. The query language of Ariel is a subset of POSTQUEL, extended with a new production-rule sub-language. Ariel rules can have conditions based on a mix of selections, joins, events, and transitions. For testing rule conditions, Ariel makes use of a discrimination network composed of a special data structure for testing single-relation selection conditions efficiently, and a modified version of the TREAT algorithm [196], called A-TREAT for testing join conditions [119]. Ariel work has also focused on optimization of discrimination network structure to provide good performance given the database update pattern, size of relations and memory nodes, join relationships in the rule condition, etc. [114]. The system also provides support for streamlined development of reliable, recoverable applications that can receive data from database triggers asynchronously [117]. The Ariel communication scheme between the active database and the application has the potential to deal with applications such as safety and integrity alert monitors, financial trading analysis programs, and command and control systems.

The Ariel system has raised the need for asynchronous interaction between application and ADBMS, as a further research direction [118, 116]. The asynchronous interaction of application and ADBMS has lead to research in cooperating application model design. It also raised issues for

solving the "correctness" issues of such asynchronous applications using extended transaction models and concurrency control schemes, as we have reviewed in this paper.

7.4 AIS

AIS (Active Information System) is an active extension on top of Oracle and has been developed at the University of Oldenburg, Germany. A prototype of the system has been built on which several applications are running. This project provides a tool-box for browsing, designing, editing, simulation, debugging of production rules, and explanation of rule results. The interactive rule debugging tool [18] supports simulation of the rules by generating realistic and relevant event sequences derived partially from the rule definitions themselves. Simulation of basic and composite events, event parameters, and incremental partial extensions of given situations can also be taken care of by the rule debugging tool.

The AIS system has also lead to the research in designing active repositories [128] and use of ADBMS for workflow management [129]. The idea of active repositories that partially automate scheduling and controlling of the activities described within a process model was introduced in [18]. It is based on active database technology that supports detection of events and triggering of corresponding actions. Events correspond to state changes in the repository or are raised by external components, e.g. a clock or CASE tool. Actions manipulate repository, trigger CASE tools, signal external systems or notify the user. The AIS system has also pointed out the necessity of extended transaction models for mapping the action-subaction hierarchy, to define "correctness" of such action-subaction hierarchy execution, as a further research issue.

7.5 ETM

ETM (Event Trigger Mechanism) is an extension to the DAMASKUS DBMS (FZI at the University of Karlsruhe, Germany). ETM [147] was developed to maintain higher consistency concepts in design applications (e.g. VLSI design). Rules watch over constraints and react by invoking suitable programs. ETM research also focuses on transaction management in the presence of rule execution in general production systems [60]. It classifies different inference mechanisms and how they interfere with the notion of transactions. The JOKER system, which has been developed as a part of the ETM project, supports both two phase locking and optimistic concurrency control for synchronization of complex transaction.

8 Active OODBMS Projects and Current Research Directions

8.1 HiPAC

The HiPAC (High Performance Active) Database System [63, 62, 64], developed at Xerox, is one of the pioneering work on active object databases. This system focused on time constrained applications. HiPAC events are very general, and can include data manipulation language (DML) operations, schema manipulation operations, transaction events, various kinds of temporal events (absolute, relative, and periodic), and external notifications from database application programs. These *primitive* events can be combined into complex events by disjunction or sequence operators. HiPAC conditions are queries in an object oriented DML. The queries can utilize information about the events as well as the state of the database. HiPAC actions can be database operations or messages to application programs, such as a request for some type of services.

HiPAC rules have five parts: event, condition, action, timing constraints, and contingency plans. The rules can have following two types of couplings among events, conditions and actions.

1. *E-C Coupling.* This coupling specifies when and how the condition will be evaluated relative to the occurrence of the event.

2. *C-A coupling.* The C-A coupling specifies when the action is executed relative to the evaluation of the condition.

Couplings in HiPAC affect database transactions. For both E-C and C-A coupling the immediate, deferred and detached modes of rule evaluation are supported. The execution model of HiPAC [124] extends a nested transaction model [156] with a variety of coupling modes that may be specified between the event part and the condition part of a rule, and between the condition part and the action part. These coupling modes provide flexibility and user control in defining the behavior of an active DBMS, instead of always executing sequentially within the triggering transaction.

Several techniques have been proposed for efficiently evaluating conditions. These includes the evaluation of multiple conditions simultaneously, incremental evaluation [167], the maintenance of derived data, and exploiting knowledge of the action parts of the rules [46]. HiPAC also investigates techniques for scheduling transactions to satisfy both the consistency constraints of execution model and timing constraints [63, 62]. It requires integration of database concurrency control techniques and time-constrained scheduling techniques developed for real-time operating systems.

An active DBMS architecture that reflects the modularity and flexibility inherent in ECA rules has also been proposed. This architecture extends the functions of the trans-

action manager and object manager components of a passive DBMS, and adds an interface for event detector components (which signal the occurrence of events of various types), a condition monitor (which accepts conditions to evaluate and optimize their evaluation), and a rule manager (which controls the processing of the rules). Important architectural issues include the interaction of active DBMS components, especially the transaction manager with the operating system and mapping of the architecture into distributed and multiprocessor architectures.

The HiPAC project has given considerable attention to long-running transactions [66] as a further research direction. With a variety of coupling modes and the ability to trigger one transaction based on the effects of another, cooperating sets of transactions can be used to support applications such as hospital management systems or office information systems. Long running transactions can respond to each other and may be used to express office protocols, such as the routing of paper work for which approvals of different managers are required. A model of long-running transactions suited to an active database environment has also been investigated in HiPAC [66].

8.2 Ode

Ode [6, 99] was developed at AT&T Bell Laboratories. It is primarily an object-oriented DBMS with activity extensions. Ode's database programming language O++ extends the object definition of C++ by supporting persistent objects. Ode contains facilities to define constraints, and boolean conditions with the class specifications [6]. Ode also allows definition of triggers that monitor the database for occurrence of some conditions. When these conditions become true, the associated action is executed. The semantics of composite events are defined as a mapping to the event history, while the composite event detection is implemented by a finite automata [101, 103]. A prototype named COMPOSE for composite event detection based on finite automata has been implemented [100]. Integrity maintenance in Ode takes care of referential integrity, uniqueness Integrity, and relational integrity [127].

8.3 Sentinel

Sentinel combines activity and object orientation in a DBMS. It treats rules and events as first class objects, and extends the object concept by an asynchronous interface for the generation and notification of events [11, 47]. Snoop [53], a language for event specification in Sentinel, has been designed to specify the primitive and composite events based on a global history log and with different parameter contexts [48]. The concept of *inter-transactional* and *intra-transactional* events and their detection has also been taken care of [47] in Sentinel. Although Sentinel has adopted the execution model of HiPAC [124], the "correctness" issues of transaction execution in the presence of inter-transactional events and deferred mode rules have not been considered. Sentinel has also been used for cooperative problem solving [52] where events and context information of long running activities have to be persistent. Complex cooperation dependencies are defined by the algebraic expressions involving events and contexts. The active database is used as persistent blackboards for basic and complex event detection and notifies the application layer.

8.4 REACH

The active OODBMS REACH (REal time ACtive and Heterogeneous mediator system) [33] supports complex applications by three types of rules, namely (i) integrity rules, (ii)access control rules, and (iii) transaction modeling rules. Application semantics are modeled by integrity rules which allow controlled inconsistency and use time constraints for specification when an integrity constraint has to be observed within the system. The access control rules and transaction modeling rules provides a more open OODBMS functionality. The transaction modeling rules are specifically used for designing workflow management applications [32]. The REACH system also supports soft and hard real time capabilities in an active DBMS environment [24]. The execution model of REACH is similar to HiPAC, but it introduces two additional coupling modes for handling irreversible actions and contingency plans [32]. The REACH prototype introduces tightly integrated event detection and rule execution mechanism [31]. This prototype focused on a clean integration of database management and active capabilities by extending Open OODB [197], an extensible OODBMS that supports low level event detection as a basic mechanism for providing extensibility.

8.5 SAMOS

SAMOS [98, 97] is an active OODBMS developed at University of Zurich, Switzerland. While aiming at HiPAC functionality, it has a richer language comparable to Ode. A prototype of SAMOS has been implemented on top of ObjectStore. SAMOS uses colored Petri nets for complex event detection [93]. The SAMOS Petri nets handle typed tokens and places for representing event parameters. Instead of using a finite automaton for complex event detection as in Ode, the typed Petri nets provide an integrated solution for handling events and event parameters as well [96].

8.6 ADAM

ADAM is an object-oriented DBMS project with prototype implementation using PROLOG. Active extensions of ADAM treat method invocations as events. ADAM manages rules by integration of active rules as first class objects [71]. Rules are managed by methods attached to them. On a metalevel, rule classes specify different rule properties and semantics [69]. The execution model of ADAM

can adopt several semantics by using inheritance for rule classes [69]. The rules also can be optimized by using class-based rule indexes [71]. An application of ADAM uses active rules to support dynamic displays [70].

9 Conclusion

In this paper, we have reviewed some of the recent developments made in the areas of active databases, specifically in the area of application modeling and concurrency control. Some of the application systems that have been developed on top of these active RDBMS and active ODBMS are first discussed. It has been observed that though there are a plethora of advanced applications that have been developed on top of these ADBMSs, there is still need for a suitable application modeling technique. The concurrency control scheme for these advanced applications running on ADBMSs has to support cooperative and long duration transactions. The reported research works on extended transaction models and the way these models have been adapted in existing ADBMSs/AODBMSs to support execution model of rules have been investigated. The major active database projects that have been carried out so far, in both active RDBMS and active ODBMS, have also been reviewed and the current research directions motivated by these projects are discussed.

References

[1] Codasyl data description language. *Journal of Development, NBS Handbook 113*, June 1973.

[2] *Interbase DDL Reference Manual, Version 3.0.* Interbase Software Corp., Bedford, Mass., 1990.

[3] *Transact-SQL User's Guide, Release 4.2.* Sybase Inc., Berkley, Callifornia, May 1990.

[4] Special Issue on Unconventional Transaction Management. *Bulletin of the IEEE Technical Committee on Data Engineering*, 14(1), March 1991.

[5] *Transaction Processing: Concepts and Techniques.* Morgan Kauffman, 1993.

[6] R. Agrawal and N. H. Gehani. Ode (Object Database and Environment): The Language and the Data Model. In *Proc. ACM-SIGMOD Intl. Conf. on Management of Data*, June 1989.

[7] Rakesh Agrawal, Roberta Cochrane, and Bruce Lindsay. On Maintaining Priorities in a Production Rule System. In *Proceedings of 17th International Conference on Very Large Databases*, pages 479–487, August 1991.

[8] A. Aiken, J. M. Hellerstein, and J. Widom. Static Analysis Techniqes for Predicting the Behaviour of Active Database Rules. *ACM Transactions on Database Systems*, 20(1):3–41, 1995.

[9] A. Aiken, J. Widom, and J. M. Hellerstein. Behaviour of Database Production Rules: Termination, Confluence and Observable Determinism. In *Proceedings 1992 ACM SIGMOD International Conference on Management of Data*, June 1992.

[10] V. Antonellis and B. Zonta. Modeling Events in Database Applications Design. In *Proc. 7th Intl. Conf. on Databases, Carmon, France*, pages 23–31, 1981.

[11] E. Anwar, L. Maugis, and S. Chakravarthy. A New Perspective for Rule Support for Object Oriented Database. In *Proc. ACM-SIGMOD International Conference on Management of Data*, pages 99–108, May 1993.

[12] P. Attie, M. Singh, M. Rusinkiewicz, and A. Sheth. Specifiying and enforcing intertask dependencies. In *Proceedings of 19th International Conference on Very Large Databases*, August 1993.

[13] J. Bailey, G. Dong, and K. Ramamohanarao. Decidability and Undecidability Results for the termination problem of active database rules. In *Proceedings of the 17th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 264–273, Seattle, Washington, 1998.

[14] E. Baralis, S. Ceri, and S. Paraboschi. Modularization Techniques for Active Rules Design. *ACM Transactions on Database Systems*, 21(1):1–29, March 1996.

[15] F. Barbic and B. Pernici. Time Modeling in Office Information Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 51–62, May 1985.

[16] J. Barron. Dialogue and Process Design for Interactive Information Systems using TAXIS. In *Proceedings of the Conference on Office Information Systems*, pages 12–20, June 1982.

[17] C. Beeri and T. Milo. A Model for Active Object Oriented Database. In *Proc. 17th Int'l. Conf. on Very Large Databases*, pages 337–349, September 1991.

[18] H. Behrends. Simulation-based Debugging of Active Databases. In *Proceedings of 4th Int'l. Workshop on Research Isssues in Data Engineering (RIDE-ADS'94)*. IEEE Press, February 1994.

[19] M. Berndtsson, S. Chakravarthy, and B. Lings. Extending Database Support for Coordination Among Agents. *International Journal on Cooperative Information Systems*, 6(3-4):315–339, 1997.

[20] P. A. Bernstein and N. Goodman. Concurrency control in distributed database systems. *ACM Computing Surveys*, 13(2):185–221, June 1981.

[21] P.A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems.* Addison Wesley, 1987.

[22] J. Blakeley, P. Larson, and F. Tompa. Efficiently Updating Materialized Views. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data*, pages 61–71, May 1986.

[23] D. G. Bobrow and M. Stefik. *The Loops Manual.* Intelligent Systems Laboratory, Xerox Corporation, 1983.

[24] H. Branding and A. Buchmann. On providing soft and hard real-time capabilities in an active dbms. In *Proceedings of 1995 ACM International Workshop on Active and Real-Time Database Systems, ARTDB-95*, Workshop in Computing. Springer, June 1995.

[25] D.A. Brant and D.P. Miranker. Index Support for Rule Activation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 42–48, May 1993.

[26] M. L. Brodie. On the Development of Data Models. In *On Conceptual Modeling, Perspective from AI, Database and Programming Language*, pages 19–47. Springer-Verlag, 1984.

[27] M. L. Brodie and D. Ridjanovic. On Design and Specification of Database Transaction. In *On Conceptual Modeling, Perspective from AI, Database and Programming Language*, pages 277–306. Springer-Verlag, 1984.

[28] L. Brownston, R. Farell, E. Kant, and N. Martin. *Programming Expert System in OPS5: An Introduction to Rule-Based Programming.* Addison Wesley, 1985.

[29] A. Buchmann, M. Hornick, E. Markatos, and C. Chronaki. Specification of a Transaction Mechanism for a Distributed Active Object System. In *Proceedings of the OOPSLA/ECOOP 90 Wrokshop on Transactions and Objects*, pages 1–9, October 1990.

[30] A. Buchmann, M. T. Ozsu, M. Hornick, D. Georgakapoulos, and F. A. Manola. A Transaction Model for Acitive Distributed Object Systems. In A. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, pages 123–158. Morgan Kauffman, 1991.

[31] A. Buchmann, J. Zimmerman, J. Blakey, and D. Wells. Building an Integrated Active Object Oriented Database Management Systems : Requirements, Architecture and Design Decisions. In *Proc. of 11th Intl. Conf. on Data Engineering*, 1995.

[32] A. P. Buchmann, H. Branding, T. Kudrass, and J. Zimmerman. Rules in an Open System : The REACH Rule System. In N. Paton and M. Williams, editors, *Rules in Database System*, Workshop in Computing. Springer, September 1993.

[33] A.P. Buchmann, H. Branding, T. Kudrass, and J. Zimmerman. REACH: A REal-time, ACtive and Heterogenous mediator sytem. *IEEE Bulletin of the Technical Committe on Data Engineering*, 15(4), December 1992.

[34] P. Buneman and E. Clemens. Efficiently Monitoring Relational Databases. *ACM Transactions on Database Systems*, 4(3):368–382, September 1979.

[35] C. Bussler and S. Jablonski. Implementing Agent Coordination for Workflow Management Systems using Active Database System. In *IEEE RIDE 4th Int'l. Workshop in Research in Isssues in Data Engineering*, pages 53–61. IEEE Press, 1994.

[36] Michael J. Carey, Rajiv Jauhari, and Miron Livny. On Transaction Boundaries in Active Databases: A Performance Perspective. *IEEE Transactions on Knowledge and Data Engineering*, 3(3):320–336, 1991.

[37] S. Ceri, P. Fraternali, S. Paraboschi, and L. Tanca. Constraint Enforcement through Production Rules: Putting Active Databases to work. *IEEE Bulletin of the Technical Committe on Data Engineering*, 15(4), December 1992.

[38] S. Ceri, P. Fraternali, S. Paraboschi, and L. Tanca. Automatic Generation of Production Rules for Integrity Maintenance. *ACM Transactions on Database Systems*, 19(3):492–535, September 1994.

[39] S. Ceri, P. Fraternali, S. Paraboschi, and L. Tanca. *Active Rule Management in Chimera*. Morgan Kauffman, 1996.

[40] S. Ceri and R. Manthey. Consolidated Specification of Chimera, the Conceptual Interface of Idea. Technical Report IDEA.DD.2P.004, Politechnico di Milano, Milan, Italy, June 1993.

[41] S. Ceri and J. Widom. Deriving Production Rules for Constraint Maintenance. In *Proceedings of 16th International Conference on Very Large Databases*, pages 566–577, August 1990.

[42] S. Ceri and J. Widom. Deriving Production Rules for Incremental View Maintenance. In *Proceedings of 17th International Conference on Very Large Databases*, pages 577–589, September 1991.

[43] S. Ceri and J. Widom. Production Rules in Parallel and Distributed Database Environments. In *Proceedings of 18th International Conference on Very Large Databases*, pages 339–351, August 1992.

[44] S. Ceri and J. Widom. Managing Semantic Heterogenity with Production Rules and Persistent Queues. In *Proceedings of Nineteenth International Conference on Very Large Databases*, pages 108–119, August 1993.

[45] S. Ceri and J. Widom. Deriving Production Rules for Deductive Data. *Information Systems*, 1994.

[46] S. Chakravarthy. Rule management and evaluation: An active dbms perspective. *ACM SIGMOD Record*, 18(3):20–28, September 1989.

[47] S. Chakravarthy, E. Anwar, and L. Maugis. Design and Implementation of Active Capability for an Object-Oriented Database. Technical Report UF-CIS-TR-93-001, University of Florida, Gainesville, USA, January 1993.

[48] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S.K.Kim. Composite Events for Active databases: Semantics, Context and Detection. In *Proc. 20th Int'l. Conf. on Very Large Databases*, pages 606–617, September 1994.

[49] S. Chakravarthy, V. Krishnaprasad, Z. Tamizuddin, and R. H. Badani. ECA Rule Integration into an OODBMS: Architecture and Implementation. In *Tech. Report UF-CIS-TR-94-023, Dept. of C.I.S., Univ. of Florida, Guinesville, Canada*, 1994.

[50] S. Chakravarthy, V. Krishnaprasad, Z. Tamizuddin, and R. H. Badani. ECA Rule Integration into an OODBMS: Architecture and Implementation. In *Proc. 11th Int'l. Conf. on Data engineering, Taipei, Taiwan*, March 1995.

[51] S. Chakravarty and E. Anwar. Exploiting Active Database Paradigm for Supporting Flexible Transaction Models. *Tech. Report UF-CIS-TR-95-026, University of Florida, Gainsvelle*, April 1995.

[52] S. Chakravarty, E. Karlapalem, S. B. Navathe, and A. Tanaka. Database Supported Cooperative Problem Solving. *Tech. Report UF-CIS-TR-92-046, University of Florida, Gainsvelle*, December 1992.

[53] S. Chakravarty and D. Mishra. Snoop: An Expressive Event Specification Language for Active Databases. *Data and Knowledge Engineering*, (14):1–26, 1994.

[54] R. Chandra and A. Segev. Active Databases for Financial Applications. In *IEEE RIDE Proceedings of 4th Int'l. Workshop on Research Isssues in Data Engineering*. IEEE Press, February 1994.

[55] P. K. Chrysanthis. *ACTA, A Framework for Modeling and Reasoning about Extended Transactions*. PhD thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, September 1991.

[56] P.K. Chrysanthis and K . Ramamritham. ACTA: A Framework for Specifying and Reasoning about Transaction Structure and Behaviour. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 194–203, May 1990.

[57] P.K. Chrysanthis and K . Ramamritham. ACTA: A Framework for Specifying and Reasoning about Transaction Structure and Behaviour. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 194–203, May 1990.

[58] P.K. Chrysanthis and K . Ramamritham. ACTA: The SAGA continues. In *Database Transaction Models for Advanced Applications*. A. Elmagarmid(ed.), Morgan Kauffman, 1992.

[59] C. Collet and T. Coupaye. NAOS : Efficient and Modular Reactive Capabilities in an OODBMS. In *Proc. 20th Int'l. Conf. on Very Large Databases, Santiago, Chille*, September 1994.

[60] C. Danner and M. Ranft. Transaction management to support rule based database application. In N. Paton and M. Williams, editors, *Rules in Database System*, Workshop in Computing, pages 143–159. Springer, September 1993.

[61] M. Darnovsky and J. Bowman. *TRANSACT-SQL User's Guide, Document 3231-2.1*. Sybase Inc., Berkley, Callifornia, 1987.

[62] U. Dayal. Active Database Management Systems. In *Proc. 3rd Intl. Conf. of Data and Knowledge Bases*, pages 150–169, 1988.

[63] U. Dayal, B. Blaustein, A. Buchmann, and S. chakravarthy. The HiPAC project: Combining active databases and timing constraints. *ACM SIGMOD Record*, 17(1):51–70, March 1988.

[64] U. Dayal, A. Buchmann, and D. McCarthy. Rules are objects too: A knowledge model for an active, object-oriented database management system. In *Proc. 2nd Intl. Workshop on Object-Oriented Database Systems, Ebernburg, Germany*, September 1988.

[65] U. Dayal, M. Hsu, and R. Ladlin. Organizing Long Running Activities with Triggers and Transactions. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data, Atlantic City*, May 1990.

[66] U. Dayal, M.Hsu, and R. Ladin. A Transactional Model for Long-running Activities. In *Proc. 17th Int'l. Conf. on Very Large Data Bases, Barcelona*, September 1991.

[67] L.M.L. Delcambre and J.N. Etheredge. The Relational Production Language: A Production Language for Relational Databases. In *Expert Database Systems - Proc. Second International Conference*, pages 333–351. Benjamin/Cummings, 1989.

[68] H.M. Dewan, D. Ohsie, S.J. Stolfo, O. Wolfson, and S. Da Silva. Incremental Database Rule Processing in PARADISER. *Journal of Intelligent Information Systems*, 1992.

[69] O. Diaz and S. M. Embury. Generating active rules from high-level specifications. In *Proceedings of the 10th British National Conference on Databases*, 1992.

[70] O. Diaz, A. Jaime, N. Paton, and G. Al-Quaimari. Supporting Dynamic Displays using Active Rules. *ACM SIGMOD Record*, 23(1):21–26, March 1994.

[71] O. Diaz, N. Paton, and P. Gray. Rule management in object-oriented databases : A uniform apporach. In *Proceedings of 17th International Conference on Very Large Databases*, September 1991.

[72] K. R. Dittrich, S. Gatziu, and A. Geppert. The Active Database Management System Manifesto: A Rulebase of ADBMS Features. In T. Sellis, editor, *Proc. of the 2nd Workshop on Rules in Databases (RIDS), Lecture Notes in Computer Science -Vol-985*. Springer, September 1995.

[73] Lyman Do and Pamela Drew. Active Database Management of Global Data Integrity Constraints in Heterogenous Database Environments. In *Proc. of 11th Intl. Conf. on Data Engineering*, pages 99–108, March 1995.

[74] A. Dori, D. Gal, and O. Etzion. Temporal Active Databases: A Key to Computer Integrated Manufacturing. *ACM SIGOIS Bulletin*, 9(2):89–104, 1996.

[75] A. Elmagarmid, editor. *Database Transaction Models for Advanced Applications*. Morgan Kauffman, 1991.

[76] S. Embury and P. M. D. Gray. Database Internal Applications. In *Active Rules in Database Systems*, Monographs in Computer Science, chapter 19, pages 339–366. Springer, 1999.

[77] K. Eswaran, J. Gray, R. Lorie, and I. Traiger. The Notion of Consistency and Predicate Locks in a Database Systems. *Communications of the ACM*, 19(11):624–633, November 1976.

[78] K.P. Eswaran and D.D. Chamberlain. Functional Specification of a Subsystem for Data Base Integrity. In *Proceedings of 1st International Conference on Very Large Databases*, September 1975.

[79] K.P. Eswaran and D.D. Chamberlain. Specification, Implementation and Interactions of a Trigger Subsystem in an Integrated Data Base System. In *IBM Research Report RJ1820*, August 1976.

[80] M. Stonebraker et. al. A Rules System for a Relational Data Base Management System. In *Proceedings of the 2nd International Conference on Databases*, June 1982.

[81] O. Etzion. Active Handling of Incomplete or Exceptional Information in Database Systems. In *Workshop on Information Technologies and Systems*, pages 46–60, December 1991.

[82] O. Etzion. Pardes - a Data-driven oriented Active Database Model. *ACM SIGMOD Record*, 22(1), March 1993.

[83] O. Etzion. An alternative paradigm for active databases. In *IEEE RIDE Proceedings of 4th Int'l. Workshop on Research Isssues in Data Engineering*. IEEE Press, February 1994.

[84] O. Etzion, D. Dori, and S. Nof. Active Coordination of CIM Multidatabase Systems. *Intrnational Journal of CIM*, 8(2), 1995.

[85] C. L. Forgy and J. McDermott. OPS – a Domain Independent Production Systems Language. In *Proceedings of Fifth International Conference on Artificial Intelligence*, 1977.

[86] P. Fraternali and S. Paraboschi. Chimera: A Language for Designing Rule Applications. In *Active Rules in Database Systems*, Monographs in Computer Science, chapter 17, pages 309–322. Springer, 1999.

[87] M. Fugini, O. Nierstrasz, and B. Percini. Application Development through Reuse: The ITHACA Tools Environments. *ACM SIGOIS Bulletin*, 13(2):38–47, August 1992.

[88] H. Garcia-Molina, , and K. Salem. SAGAS. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 249–259, May 1987.

[89] H. Garcia-Molina. Using semantic knowledge for transaction processing in distributed databases. *ACM Transactions on Database Systems*, 8(2), June 1983.

[90] H. Garcia-Molina, D. Gawlick, J. Klein, K. Kleinser, and K. Salem. Coordinating Multi-Transaction Activities. In *Tech. report CS-TR-247-90, Dept. of Computer Science, Princetion University*, February 1990.

[91] H. Garcia-Molina, D. Gawlick, J. Klein, K. Kleinser, and K. Salem. Modeling Long-Running Activities as Nested Sagas. *Bulleitn of the IEEE Technical Committee on Data engineering*, 14(1):14–18, March 1991.

[92] S. Gatziu. *Events in an Active Object-Oriented Database Systems*. PhD thesis, Institut fur Informatik, Universtat Zurich, Switzerland, 1995.

[93] S. Gatziu and K. R. Dittrich. Events in an Active Object-Oriented System. In N. Paton and M. Williams, editors, *Rules in Database System*, Workshop in Computing, pages 127–142. Springer, September 1993.

[94] S. Gatziu and K. R. Dittrich. Events in an Object-Oriented Database Systems. In *Proc. 1st Intl. Conf. on Rules in Database Systems*, 1993.

[95] S. Gatziu and K. R. Dittrich. Detecting Composite Events in Active Database Systems using Petri-Nets. In *IEEE RIDE Proceedings of 4th Int'l. Workshop in Research in Isssues in Data Engineering*. IEEE Press, February 1994.

[96] S. Gatziu and K. R. Dittrich. Detecting composite events in active database systems using petri nets. In *Proceedings of 4th Int'l. Workshop on Research Isssues in Data Engineering (RIDE-ADS'94)*. IEEE Press, February 1994.

[97] S. Gatziu, A. Geppert, and K.R. Dittrich. Integrating active concepts into an object-oriented database. In *Proc. of the 3rd International Workshop on Database Programming Languages*, August 1991.

[98] Stella Gatziu and K. R. Dittrich. SAMOS: An Active Object Oriented Database System. *IEEE Quarterly Bulletin on Data Engineering*, 15((1-4)), December 1992.

[99] N. Gehani and H. V. Jagadish. Ode as an Active Database: Constraints and Triggers. In *Proc. 17th Int'l. Conf. on Very Large Databases, Barcelona*, September 1991.

[100] N. Gehani, H. V. Jagadish, and O. Shmueli. *COMPOSE: A System for Composite Event Specification and Detection*, chapter Chapter 1. Springer, 1993.

[101] N. Gehani, H. V. Jagadish, and O. Shumeli. Composite Event Specification in Active Databases: Model and Implementation. In *Proceedings of 18th International Conference on Very Large Databases*, August 1992.

[102] N. H. Gehani and H. V. Jagadish. Active Database Facilities in ODE. In J. Widom and S. Ceri, editors, *Active Database Systems - Triggers and Rules for Advanced Database Processing*. Morgan Kauffman Pulishers Inc., 1996.

[103] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Events Specification in an Active Object-Oriented Database. In *Proc. ACM-SIGMOD Intl. Conf. on Management of Data*, pages 81–90, June 1992.

[104] A. Geppert and K. R. Dittrich. Rulebased Implementation of Transaction Model Specifications. In N. Paton and M. Williams, editors, *Rules in Database System*, Workshop in Computing, pages 127–142. Springer, September 1993.

[105] A. Geppert, M. Kradolfer, and D. Tombros. Realization of Cooperatvie Agents using an Active Object-oriented Database Management Systems. In T. Sellis, editor, *Proc. of the 2nd International Workshop on Rules in Databases System, (RIDS), LNCS-985*, pages 327–341. Springer, September 1995.

[106] M. Gerth and U. W. Lipeck. Deriving Integrity Maintaining Triggers from Transition Graphs. In *Proceedings of 9th International Conference on Data Engineering*, pages 22–29, April 1993.

[107] M. Gertz. Specifying Reactive Intcgrity Control for Active Databases. In *IEEE RIDE Proceedings of 4th Int'l. Workshop in Research in Isssues in Data Engineering*. IEEE Press, February 1994.

[108] S. Ghandeharizadeh, R. Hull, and D. Jacobs. On Implementing a Language for Specifying Active Database Execution Models. In *Proceedings of 19th International Conference on Very Large Databases*, August 1993.

[109] J. Gray. The Transaction Concept: Virtues and Limitations. In *Proceedings of Seventh International Conference on Very Large Databases*, pages 144–154, September 1981.

[110] J.N. Gray, R. A. Lorie, G. R. Putzulo, and I. L. Traiger. Granularity of locks and degrees of consistency in a shared database. In M. Stonebraker, editor, *Readings in Database System*, pages 94–121. Morgan Kauffman, 1988.

[111] L. Haas, W. Chang, G. Lohman, J. McPherson, P. Wilms, B. Lindsay, H. Pirahesh, M. Carey, and E. Shekita. Starburst mid-flight: as the dust clears. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):143–160, March 1990.

[112] T. Haerder and A. Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317, December 1983.

[113] E. Hanson. A Performance Analysis of View Materialization Strategies. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 440–453, May 1987.

[114] E. Hanson. Gator: A Generalized Discrimination Network for Production Rule Matching. In *Proceedings of IJCAI Workshop on Production Systems and their Innovative Applications*, August 1993.

[115] E. N. Hanson. An initial report on the design of Ariel: a DBMS with an integrated production rulle system. *ACM SIGMOD Record*, 18(3):12–19, September 1989.

[116] E. N. Hanson, C. Carnes, L. Huang, M. Konyala, L. Noronha, S. Parthasarathy, J. B. Park, and A. Vernon. Scalable Trigger Processing. In *Proceedings of the 15th International Conference on Data Engineering*, pages 266–275. IEEE Computer Society Press, 1999.

[117] E. N. Hanson, R. Dastur, and V. Ramaswamy. An Architecture for Recoverable Interaction between Application and Active Databases. Technical Report CIS-TR-93-024, University of Florida, Gainesville, USA, July 1993. extended abstract.

[118] E. N. Hanson and S. Khosla. An Introduction to the TriggerMan Asynchronous Trigger Processor. In *Proceedings of the 3rd International Workshop on Rules in Database Systems*, volume 1312 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 1997.

[119] Eric N. Hanson. Rule Condition Testing and Action Execution in Ariel. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, June 1992.

[120] Eric N. Hanson. The Design and Implementation of the Ariel Active Database Rule System. *IEEE Transactions on Knowledge and Data Engineering*, 8(1):157–172, February 1996.

[121] T. Harder and K. Rothermal. Concurrecny Control Issues in Nested Transactions. *VLDB Journal*, 2(1):39–74, 1993.

[122] H. Herbst. A Meta-Model for Business Rules in System Analysis. In J. Iivari, K. Lyytinen, and M. Rossi, editors, *Proc. of the 7th Intl. Conf. on Advanced Information Systems Engineering (CAiSE95), LNCS 392*. Springer, June 1995.

[123] C. Hewitt, P. Bishop, and R. Steiger. A universal modular ACTOR fromalism for artificial intelligence. In *Proceedings of 3rd International Joint Conference on Artificial Intelligence*, pages 235–245, 1973.

[124] M. Hsu, R. Ladin, and D.R. McCarthy. An Execution Model for Active Database Management Systems. In *Proc. 3rd Intl. Conf. of Data and Knowledge Bases*, pages 171–179, June 1988.

[125] S. Hudson and R. King. CACTIS: A Database System for Specifying Functionally-Defined Data. In *Proc. of 1st Int'l. Workshop on Object-Oriented Database Systems*, September 1986.

[126] R. Hull and D. Jacobs. Language Constructs for Programming in Active Databases. In *Proc. 17th Int'l. Conf. on Very Large Data Bases, Barcelona*, September 1991.

[127] H. V. Jagadish and X. Qian. Integrity Maintenance in an Object-Oriented Database. In *Proceedings of 18th International Conference on Very Large Databases*, September 1992.

[128] H. Jasper. Active Datbases for Active Repositories. In *Proceedings of 9th International Conference on Data Engineering*, pages 22–29, April 1993.

[129] H. Jasper, O. Zukunft, and H. Behrends. Time Issues in Advanced Workflow Manangement Applications of Active Databases. In *Proceedings of 1995 ACM International Workshop on Active and Real-Time Database Systems, ARTDB-95*, Workshop in Computing. Springer, June 1995.

[130] Rammohanrao Jawadi and Stanley Y. W. Su. Incorporating Flexible and Expressive Rule Control in a Graph-based Transaction Framework. Technical Report UF-CIS-TR-94-030, University of Florida, Gainesville, USA, 1994.

[131] P. Kangsabanik, R. Mall, and A. K. Majumdar. Modeling Applications in Active Object Oriented Database Management System. In *4th International Conference on Object Oriented Information Systems (OOIS-96)*. Springer-Verlag, 1996.

[132] P. Kangsabanik, R. Mall, and A. K. Majumdar. Modeling Long-Duration and Cooperative Transactios in Active Object Oriented Database Management System. In Sushi Jajodia and Eliza Bertino, editors, *Proceedings of the International Workshop on Advanced Trnasaction Models and Architectures - ATMA96 (in conjinction with VLDB-96)*, pages 96–106, 1996.

[133] P. Kangsabanik, R. Mall, and A. K. Majumdar. Concurrency Control of Nested Cooperativce Transactions in Active DBMS. In *IEEE 4th International Conference on High Peformance Computing (HiPC-97)*, pages 1–8. IEEE Press, December 1997.

[134] P. Kangsabanik, R. Mall, and A. K. Majumdar. A Technique for Modeling Applications in Active Object Oriented Database Management System. *Information Sciences - An International Journal*, 102(1-4):67–104, 1997.

[135] P. Kangsabanik, R. Mall, and A.K. Majumdar. Towards a Model of Concurrency for Executing Cooperative Transactions in Acitve OODBMS. In *Proceedings of the National Seminar on Theoritical Computer Scinece (NSTCS-7)*, June 1997.

[136] G. Kappel, S. Rausch-Schott, W. Retschitzegger, and M. Sakkinen. Multi-Parent Subtransactions Covering the Transactional Needs of Composite Events. In E Bertino, S. Jajodia, and L. Kerschberg, editors, *Proc. of International Workshop on Advanced Transaction Models and Architectures (ATMA-96)*, pages 269–282, August 1996.

[137] G. Kappel, S. Rausch-Schott, W. Retschitzegger, and M. Sakkinen. A Transaction Model for Handling Composite Events. In *Proc. of the 3rd International Workshop on Advance in Database and Information Systems (ADBIS-96)*, pages 116–125, September 1996.

[138] G. Kappel and W. Retschitzegger. The TriGS Active Object-Oriented Database System - An Overview. *ACM SIGMOD Record*, 27(3):36–41, September 1998.

[139] R. King and D. Mcleod. A Unified Model and Methodology for Conceptual Database Design. In *On Conceptual Modeling, Perspective from AI, Database and Programming Language*, pages 313–327. Springer-Verlag, 1984.

[140] G. G. Knolmayer, H. Herbst, and M. Schlesinger. Enforcing Business Rules by the Application of Trigger Concepts. In *Proceedings of Priority Programme Informatics Resarch, Information Conference Module 1: Secure Distributed Systems*, pages 24–30, November 1994.

[141] S. Koenig and R. Paige. A Transformational Framework for Automatic Control of Derived Data. In *Proceedings of 1st International Conference on Very Large Databases*, September 1975.

[142] H. F. Korth, E. Levy, and A. Silberschatz. Compensating Transactions: A New Recovery Paradigm. In *Proceedings of Sixteenth International Conference on Very Large Databases*, pages 95–106, August 1990.

[143] H. F. Korth and G. Speegle. Formal Models of Correctness without Serializability. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 379–386, June 1988.

[144] Henry F. Korth, E. Levy, and A. Silberschatz. On Long-Duration CAD Transactions. *Information Sciences*, 46(1-2):73–107, October-November 1988.

[145] A. Koschel and Peter C. Lockemann. Distributed Events in Active Database Systems - Letting the Genie out of the Bottle. *Journal of Data and Knowledge Engineering (DKE)*, 25, 1998. Special Issue for the 25th Vol. of DKE.

[146] A. Kotz, K. Dittrich, and J. Mulle. Supporting semantic rules by a generalized event/trigger mechanisms. In *Proceedings of International Conference on Extending Database Technology*, March 1988.

[147] A. M. Kotz, K. R. Dittrich, and J. Mulle. Supporting Semantic Rules by a generalized Event/Trigger Mechanism. In J. W. Schmidt, S. Ceri, and M. Missikoff, editors, *Proc. of the Intl. Conf. on Extending Database Technology (EDBT-88), LNCS 303*, pages 76–92. Springer, 1988.

[148] K. Kulkarni, N. Mattos, and R. Cochrane. Active Database Features in SQL3. In N. W. Paton, editor, *Active Rules in Database Systems*, Monographs in Computer Science, chapter 10, pages 197–219. Springer, 1999.

[149] E. Levy, H. F. Korth, and A. Silberschatz. Compensating Transactions: A New Recovery Paradigm. In *Proceedings of Tenth Annual ACM Symposium on Principles of Distributed Computing*, August 1991.

[150] D. F. Lieuwen, N. Gehani, and R. Arlein. The ODE Active Database: Trigger Semantics and Implementation. In *Proc. of 12th Intl. Conf. on Data Engineering*, pages 412–420, 1996.

[151] B. Lindsay, L. Haas, and C. Mohan. A Snapshot Differential Refresh Algorithm. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 53–60, May 1986.

[152] P. Loucopoulos, C. Theodoulidis, and D. Pantazis. Business Rules Modelling : Conceptual Modeling and Object Oriented Specifications. In F. Van Assche, B. Moulin, and C. Rolland, editors, *IFIP WG8.1, Working Conf. on the Object-Oriented Approach in Information Systems, Quebec City, Canada*, pages 28–31. North-Holland, October 1991.

[153] N. A. Lynch. Multilevel atomicity - a new correctness for database concurrency control. *ACM Transactions on Database Systems*, 8(4):484–502, December 1983.

[154] M. Minsky. A Framework for Representing Knowledge. In P. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[155] M. Morgenstern. Active Databases as a Paradigm for Enhanced Computing Environment. In *Proceedings of 9th International Conference on Very Large Databases*, pages 34–42, September 1983.

[156] J.E.B. Moss. Nested Transactions : An Approach to Reliable Distributed Computing. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data, Atlantic City*, May 1990.

[157] A.H.H. Ngu. Transaction Modeling. In *Proc. of 5th Intl. Conf. on Data Engineering, Calliforna, USA*, pages 234–241, 1989.

[158] B. Nixon. TAXIS-84 Selected Papers. In *Tech. Report CSRG-160, Dept. of C.S.E., Univ. of Toronto, Canada*, 1984.

[159] J.J. Odell. Specifying Requirements using Rules. *Journal of Object Oriented Programming*, 6(2), 1994.

[160] M. T. Ozsu. *Transaction Models and Transaction Management in Object-Oriented Database Management Systems*. Springer Verlag, NATO ASI Series F130, 1994.

[161] C. H. Papadimitriou. *The Theory of Database Concurrency Control*. Computer Science Press, 1986.

[162] I. Petrounias and P. Loucopoulos. A Rule-Based Approach for the Design and Implementation of Information Systems. In M. Jarke, J. Bubenko, and K. Jeffery, editors, *Proc. of the 4th Intl. Conf. on Extending Database Technology (EDBT94), LNCS 779*. Springer, March 1994.

[163] P. Picouet and V. Vianu. Semantics and Expressiveness Issues in Active Databases. *Journal of Computer and System Sciences*, 57(3):325–355, December 1998.

[164] C. Pu. *Replication and Nested Transactions in the Eden Distributed Systems*. PhD thesis, University of Washington, 1986.

[165] C. Pu, G. Kaiser, and N. Hutchinson. Split-Transactions for Open-Ended Activities. In *Proc. 14th Int'l. Conf. on Very Large Databases*, pages 26–37, September 1988.

[166] L. Raschid, T. Sellis, and A. Delis. A simulation-based study on the concurrent execution of rules in a database environment. *Journal on Parallel and Distributed Computing*, 20(1):20–42, January 1994.

[167] A. Rosenthal, S. Chakravarthy, B. Blaustin, and J. Blakeley. Situation monitoring for active databases. In *Proceedings of 15th International Conference on Very Large Databases*, August 1989.

[168] N. Roussopoulos. View Indexing in Relational Databases. *ACM Transactions on Database Systems*, 7(2):258–290, June 1982.

[169] L. Rowe and M. Stonebraker. The Postgres Data Model. In *Proceedings of 13th International Conference on Very Large Databases*, September 1987.

[170] K. S. Rubin, P. McClaughry, and D. Pelligrini. Modelling Rules using Object Behaviour Analysis and Design. *Object Magazine*, June 1994.

[171] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modelling and Design*. Prentice Hall, 1991.

[172] Y. Sagin, O. Ulusoy, and S. Chakravarthy. Concurrent Rule Execution in Active Databases. *Information Systems*, 23(1):39–64, 1998.

[173] J. G. Schmolze. Guaranteeing serializable results in synchronous parallel production systems. *Journal on Parallel and Distributed Computing*, 13(4):348–365, December 1991.

[174] U. Schreier, H. Pirahesh, R. Agrawal, and C. Mohan. Alert: An Architecture for Transforming a Passive DBMS into an Active DBMS. In *Proceedings of 17th International Conference on Very Large Databases*, pages 469–478, September 1991.

[175] T. Sellis and C.C. Lin L. Raschid. Implementing Large Production Systems in a DBMS Environment: Concepts and Algorithms. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 404–412, June 1988.

[176] L. Sha. *Modular Concurrency Control and Failure Recovery - Consistency, Correctness and Optimality*. PhD thesis, Department of Computer and Electrical Engineering, Carnegie-Mellon University, 1985.

[177] M. J. V. Silva and C. R. Carlson. Conceptual Design of Active Object-Oriented Database Applications Using Multi-level Diagrams. In *Proceedings of the 10th European Conference on Object-Oriented Programming (ECCOP'96)*, volume 1098 of *Lecture Notes in Computer Science*, pages 366–397. Springer, 1996.

[178] E. Simon, J. Kiernan, and C. deMaindreville. Implementing high level active rules on top of a relational dbms. In *Proceedings of 18th International Conference on Very Large Databases*, pages 315–326, 1992.

[179] E. Simon and A. Kotz-Dittrich. Promises and Realities of Active Database Systems. In *Proceedings of 21st International Conference on Very Large Databases*, 1995.

[180] A.P. Sistla and O. Wolfson. Temporal Conditions and Integrity Constraints in Active Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 269–279, May 1995.

[181] A. Skarra and S. Zdonik. *Concurrency Control and Object-Oriented Databases*, pages 395–421. ACM Press, 1989.

[182] K. Smith and M. Winslett. Multilevel Secure Rules: Integrating the Multilevel and Active Data Models. Technical Report UIUDCS-R-92-1732, University of Illinois, Urbana-Champaign, USA, 1992.

[183] M. Stonebraker. Triggers and Inference in Database Systems. In Brodie and Mylopoulos, editors, *On Knowledge Base management Systems*. Springer-Verlag, 1986.

[184] M. Stonebraker. The integration of rule systems and database system. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):415–423, October 1992.

[185] M. Stonebraker, E. Hanson, and C.H. Hong. The Design of Postgres Rule System. In *Proc. of 3rd Intl. Conf. on Data Engineering, Callifornia, USA*, 1987.

[186] M. Stonebraker, E. Hanson, and S. Potamianos. The Postgres Rule Manager. *IEEE Transactions on Software Engineering*, 14(7):143–160, July 1988.

[187] M. Stonebraker, M. Hearst, and S. Potamianos. A Commentary on the Postgres Rule System. *ACM SIGMOD Record*, 14(7):897–909, September 1989.

[188] M. Stonebraker, A. Jhingran, J. Goh, and S. Potmianos. On Rules, Procedures, Caching and Views in Data Base Systems. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data, Atlantic City*, pages 281–290, May 1990.

[189] M. Stonebraker, L. Rowe, and M. Hiroshima. The Implementation of Postgres. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125–142, March 1990.

[190] D. Toman. Implementing Temporal Integrity Constraints using an Active DBMS. In *Proceedings of 4th Int'l. Workshop on Research Isssues in Data Engineering (RIDE-ADS'94)*, pages 87–95. IEEE Press, February 1994.

[191] A. Tsalgatidou and P. Loucopoulos. An Object-Oriented Rule-Based Approach to the Dynamic Modelling of Information Systems. In H. G. Sol and K. M. Van Hee, editors, *Proc. of the Intl. Working Conf. on Dynamic Modeling of Information Systems*, pages 165–188. North-Holland, 1991.

[192] Y. Vassilion, J. Clifford, and M. Jarke. Database Access Requirements of Knowledge-Based Systems. In Won Kim, D.S. Reiner, and D.S. Batroy, editors, *Query Processing in Database Systems*, pages 156–170. Springer-Verlag, 1985.

[193] J. Verhofstad. Recovery techniques for database systems. *ACM Computing Surveys*, 10(2):167–196, June 1978.

[194] S. vinter, K. Ramamritham, and D. Stemple. Recoverable Actions in Gutenberg. In *Proceedings of the Sixth International Conference on Distributed Computing Systems*, pages 242–249, May 1986.

[195] H. Wachter and A. Reuter. The ConTract model. In A. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, pages 219–264. Morgan Kauffman, 1991.

[196] Y.W. Wang and E.N. Hanson. A Performance Compariosn of the Rete and TREAT Algorithm for Testing Database Rule Condition. In *Proc. of Eighth Intl. Conf. on Data Engineering*, February 1992.

[197] D. L. Wells, J. A. Blakeley, and C. W. Thompson. Architecture of an Open Object-Oriented Database System. *IEEE Computer*, 25(10), October 1993.

[198] J. Widom. A Denotational Semantics for the Starburst Production Rule Language. *ACM SIGMOD Record*, 21(3), September 1992.

[199] J. Widom. The Starburst Active Database Rule Systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):583–595, 1996.

[200] J. Widom and S. Ceri, editors. *Active Database Systems: Triggers and Rules For Advanced Database Processing*. Morgan Kaufmann, 1996. ISBN 1-55860-304-2.

[201] J. Widom and Sh. Finkelstein. Set-oriented Producton Rules in Relational Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 259–270, May 1990.

[202] Jennifer Widom, R. Cochrane, and Bruce Lindsay. Implementing Set-Oriented Production Rules as an Extension to Starburst. In *Proceedings of 17th International Conference on Very Large Databases*, pages 275–285, September 1991.

[203] Jennifer Widom, J. Sheldon, and A. Finkelstein. A Syntax and Semanitcs for Set-Oriented Production Rules in Relational Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 259–270, May 1990.

[204] Ming Xiong and Krithi Ramamritham. Specification and Analysis of Transaction in Real-Time Active Databases. *Second International Workshop on Real Time Databases (RTDB)*, 1997.

[205] M. Zloof. Office-by-example: a business language that unifies data and word processing and electronic mail. *IBM Systems Journal*, 21(3):272–304, 1982.

Linear Algebra in One-Dimensional Systolic Arrays

Gregor Papa and Jurij Šilc
Computer Systems Department, "Jožef Stefan" Institute, Jamova 39, 1001 Ljubljana, Slovenia
email: gregor.papa@ijs.si, jurij.silc@ijs.si

Keywords: systolic array, QR, LU, decomposition, Gauss elimination, matrix multiplication

Edited by: Rudi Murn
Received: June 1, 1999 **Revised:** July 20, 1999 **Accepted:** April 5, 2000

Frequently used problems of linear algebra, such as the solution of linear systems, triangular decomposition and matrix multiplication, are computationally extensive. To increase the speed, those problems should be solved with systolic structures, where many processors are used concurrently to compute the result. Since two-dimensional array of processors is very consumptive, considering space and resources, it is better to use one-dimensional array of processors. This leads to the operation reallocation and causes unequal utilization of processors, but it is much easier to implement since there is only one linear array of processors.

1 Introduction

Many scientific problems can be solved by linear algebraic computations, but even some basic operations are computationally extensive. Computation time could be shortened by synchronous data processing, which is enabled through the systolic structure. Systolic solving is presented by the processor structure, where data is flowing through the net of specialized processors, which are locally connected and work synchronically. This approach has some disadvantages, while there is a lot of connections. It is difficult to monitor all processors and to read data from them. Besides, they are poorly utilized, since they mostly wait for their data to compute. It is possible to compose the structure with higher utilization, time suitability and lower complexity [3], which would remove the mentioned disadvantages. To realize that, we can merge some processors, i.e. one processor performs tasks of more processors, and we can put them into one straight array, to reduce the number of connections and to make easier access to the processors. This work presents the linearization of different matrix transformation algorithms, such as elimination, decompositions and multiplication, and also some comparisons of two-dimensional and linear arrays are given.

2 Linear system of equations

Systolic arrays can be used to solve the system of linear equations [2] in the form:

$$\mathbf{A} \cdot x = \mathbf{b}.$$

Suitable triangular systolic array for realization of Gauss elimination and various decompositions (QR and LU) [4, 9] is presented in Fig. 1. Shapes ○ and □ represent two types of processor (diagonal and inner), performing their own instructions; diagonal operations are executed in diagonal processors and inner operations are executed inside the structure. Inputs of the structure are matrix coefficients (a_{ij}) and at the end there are coefficients of the upper-triangular matrix inside the structure and the coefficients of the lower-triangular matrix on the outputs. Dotted square represents a delay τ. According to the matrix size $n \times n$ the number of required processors n^* is:

$$n^* = \frac{n(n+1)}{2}.$$

Where n diagonal processors and $(n^* - n)$ inner processors are required.

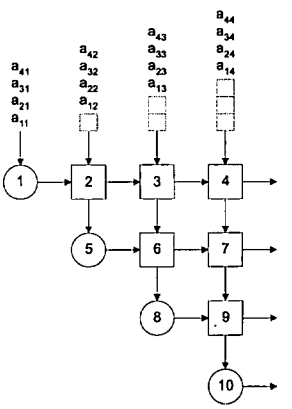
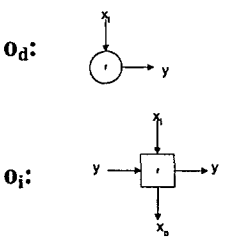


Figure 1: Triangular systolic array (n=4)

Two-dimensional array in Fig. 1 can be transformed into one-dimensional [11, 6] in several directions; horizontal linear array (Fig. 2), vertical linear array (Fig. 4), diagonal linear array (Fig. 6) and interweaved linear array (Fig. 8). Symbol ⊡ represents the processor that performs the

tasks of processors ○ and □. Next, the operations of diagonal and inner processors are presented. All mentioned operations [1] are executed in one systolic cycle (step), but of course, more cycles are needed to finish a transformation, i.e. those operation are repeated (operations present only the set of processor's instructions).

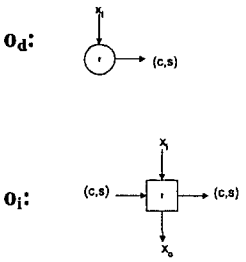
Gauss elimination [5] and LU decomposition [7]:



$$1.\ y=\frac{x_i}{r}$$
$$2.\ r=x_i$$
$$1.\ x_o=x_i-y\cdot r$$
$$2.\ r=x_i$$

In such structure there is a similarity of Gauss elimination and LU decomposition (results of LU decomposition are just transformed Gauss coefficients) [7].

QR decomposition [5]:



$$\text{if } x_i=0,\ r=0 \text{ then } c=1,\ s=0$$
$$\text{else } t=\sqrt{r^2+x_i^2},\quad c=r/t,\quad s=x_i/t,\ r=t$$
$$x_o=-s\cdot r+c\cdot x_i$$
$$r=c\cdot r+s\cdot x_i$$

Input or output (c, s) of QR decomposition will be treated as y in the following sections.

Because of the transformations the instruction sets of the processors are changed as described in the following sections.

2.1 Horizontal array

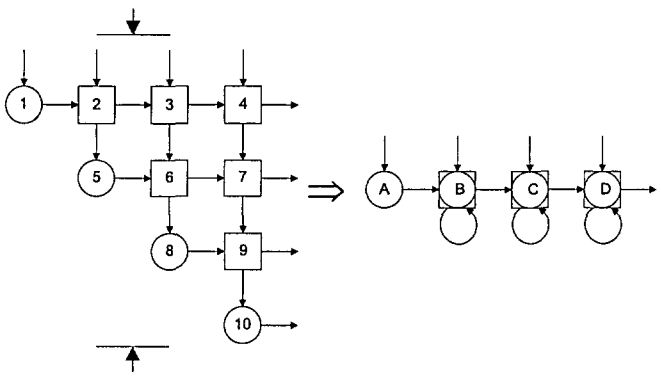


Figure 2: Transformation into horizontal array

As presented in Fig. 2, processor 1 is mapped into processor A; processors 2 and 5 into processor B, processors 3, 6 and 8 into processor C; processors 4, 7, 9 and 10 into processor D. So, processor A takes over the tasks of one processor and performs operation o_d, but processor D takes over the tasks of four processors and performs operations o_d and o_i. They work in different modes:

- mode 1: operation o_d with one input x_i,
- mode 2: operation o_i with two inputs (x_i,y),
- mode 3: operation o_i with one input y and one input from its output (x_o to x_i).

Each processor works in these modes:

- processor A always in mode 1,
- processor B in modes 2 and 1,
- processor C in modes 2, 3 and 1,
- processor D in modes 2, 3, 3 and 1,
- additional processors would work in modes 2, 3, ... 3 and 1.

Occupation of processors is presented in Table 1.

Table 1: Processor occupation in horizontal array

	A	B	C	D
1	1			
2		2		
3		1	2	
4			3	2
5	1		1	3
6		2		3
7		1	2	1
8			3	2
...				
17			1	3
18				3
19				1

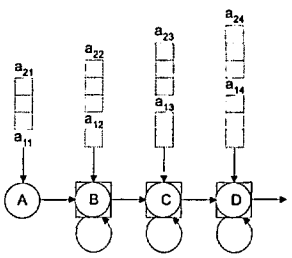


Figure 3: Data inputs in horizontal array

Input values a_{11}, a_{12}, a_{13} and a_{14} are delayed for one τ, and values a_{21}, a_{22}, a_{23} and a_{24} are delayed for $(n-1)\tau$ according to values a_{11}, a_{12}, a_{13} and a_{14}, where n is the number of processors, as presented in Fig. 3.

2.2 Vertical array

As it can be seen in Fig. 4, processors 1, 2, 3 and 4 are mapped into processor A; processors 5, 6 and 7 into processor B; processors 8 and 9 into processor C; processor 10 into processor D. Processor A is the most loaded, while processor D takes over the tasks of only one processor.

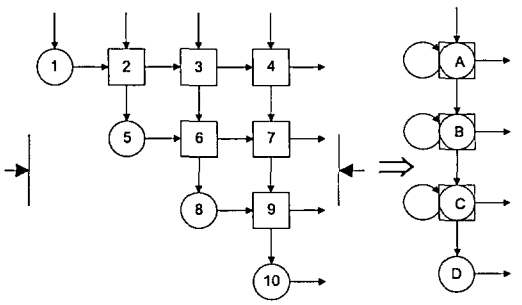


Figure 4: Transformation into vertical array

Processors A, B and C perform operations o_d and o_i, while processor D performs only operations o_d. They work in different modes:

- mode 1: operation o_d with one input x_i,
- mode 2: operation o_i with one input x_i and one input from its output (y to y).

Table 2: Processor occupation in vertical array

	A	B	C	D
1	1			
2	2			
3	2	1		
4	2	2		
5	1	2	1	
6	2		2	
7	2	1		1
8	2	2		
...				
17		2	1	
18			2	
19				1

Each processor works in these modes:

- processor A in mode 1, 2, 2 and 2,
- processor B in mode 1, 2 and 2,
- processor C in mode 1 and 2,
- processor D always in mode 1,
- additional processors would work in modes 1, 2, ...2 and 2.

Occupation of processors and their work modes are presented in Table 2. Values a_{11}, a_{12}, a_{13} and a_{14} follow each other without delay, values a_{21}, a_{22}, a_{23} and a_{24} are immediate successors of values a_{11}, a_{12}, a_{13} and a_{14}, as presented in Fig. 5.

When transformed into horizontal or vertical array, the processors' occupation and their instruction set are equal. The only difference can be noticed in data inputs.

2.3 Diagonal array

Fig. 6 presents the diagonal contraction, where processors 1, 5, 8 and 10 are mapped into processor A; processors 2, 6 and 9 into processor B; processors 3 and 7 into processor C; processor 4 into processor D. Even here the most loaded is processor A and at least processor D, but all processors

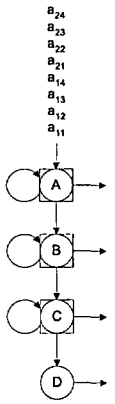


Figure 5: Data inputs in vertical array

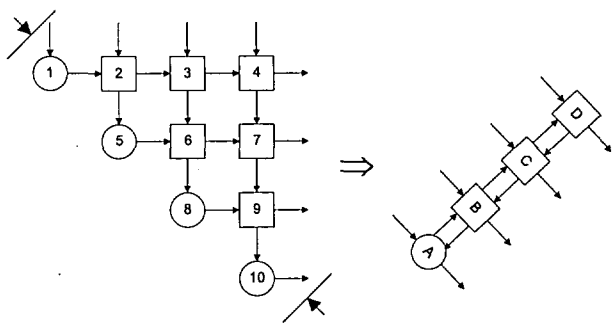


Figure 6: Transformation into diagonal array

execute only one type of operations (processor A performs only diagonal operations, the others only inner operations).

Processor occupation and their operations are presented in Table 3.

Table 3: Processor occupation in diagonal array

	A	B	C	D
1	o_d			
2	o_d	o_i		
3	o_d	o_i	o_i	
4	o_d	o_i	o_i	o_i
5	o_d	o_i	o_i	o_i
6	o_d	o_i	o_i	
7	o_d	o_i		
8	o_d			
...				
14	o_d	o_i	o_i	
15	o_d	o_i		
16	o_d			

Values a_{11}, a_{12}, a_{13} and a_{14} are one τ delayed and are followed by values a_{21}, a_{22}, a_{23} and a_{24}. Values a_{31}, a_{32}, a_{33} and a_{34}, are delayed $2(n-1)\tau$, where n is the number of processors, as presented in Fig. 7.

Contraction of the array in the direction of the other diagonal is not reasonable, while there would be too many delays and inputs/outputs on each processor.

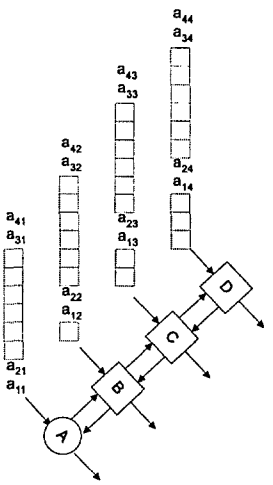


Figure 7: Data inputs in diagonal array

2.4 Processor mirroring

To decrease the number of processors and to enhance the performance of transformations, mirroring can be used. The processor can be mirrored into another processor, so that its tasks are executed while another processor would be idle otherwise. The example of processor mirroring in horizontal linear array is presented in Table 4. Processor A is mapped into processor B, and merged processor A+B executes tasks of both processors. Similarly other mirrorings can be used.

Table 4: Processor mirroring a)original array, b)array with mapped processor

a)	A	B	C	D	b)	A+B	C	D
1	1				1	1		
2		2			2	2		
3		1	2		3	1	2	
4			3	2	4		3	2
5	1		1	3	5	1	1	3
6		2		3	6	2		3
7		1	2	1	7	1	2	1
8			3	2	8		3	2
...					...			

2.5 Interweaved array

When there is an odd number of processors in the first line of the triangular array, the interweaved method can be used, as presented in Fig. 8 [11], where the isomorphic embedding of the graph is employed. Processors in Fig. 8a are mapped into processor array in Fig. 8b: processors 1, 6, 10, 13 and 15 are mapped into processor A; processors 2, 5, 7, 11 and 14 into processor B; processors 3, 4, 8, 9 and 12 into processor C. All processors (A, B, C) are evenly loaded, while each of them takes over the tasks of five processors.

The method is similar to processor mirroring, but it occupies processors almost completely and evenly. Instead

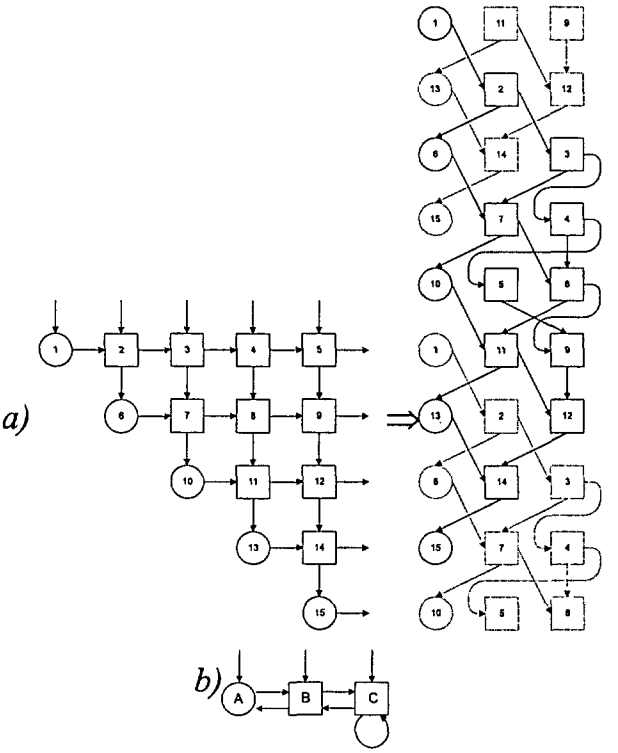


Figure 8: Transformation into interweaved array

of $n = 5$ processors only $n^* = \frac{(n+1)}{2} = 3$ are needed, which are fully utilized. Processor A performs operations o_d, while B and C perform operations o_i. Processors occupation and their operations are presented in Table 5.

Table 5: Occupation of interweaved array

	A	B	C
1	o_d		
2	o_d	o_i	
3	o_d	o_i	o_i
4	o_d	o_i	o_i
5	o_d	o_i	o_i
6	o_d	o_i	o_i
7	o_d	o_i	o_i
8	o_d	o_i	o_i
9	o_d	o_i	o_i
10	o_d	o_i	o_i
...			
27	o_d	o_i	o_i
28	o_d	o_i	
29	o_d		

Values a_{11}, a_{12}, a_{13}, a_{14} and a_{15} are delayed one τ, values a_{21}, a_{22}, a_{23}, a_{24} and a_{25} are delayed $(n^* + 1)\tau$, according to values a_{11}, a_{12}, a_{13}, a_{14} and a_{15}, as presented in Fig. 9.

3 Matrix multiplication

Systolic arrays can be also used when performing matrix multiplication [2] of the form

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} + \mathbf{C_0}.$$

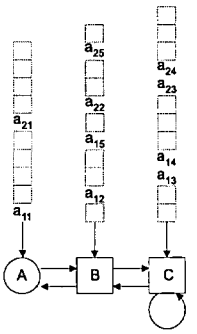


Figure 9: Data inputs in interweaved array

Square array of processors for multiplication of two square matrices is presented in Fig. 9 [8]. Inputs of the structure are coefficients (a_{ij} in b_{ij}) of the matrices and at the end of the process there are coefficients c_{ij} inside the structure. According to the matrix size $n \times n$ the number of required processors n^* is:

$$n^* = n^2.$$

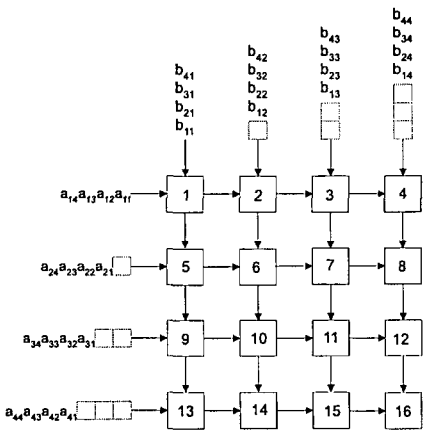
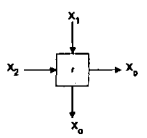


Figure 10: Square systolic array (n=4)

All processors in the square array in Fig. 10 perform the same operations [8]:



1. $x_o = x_1 \cdot x_2 + r$
2. $r = x_o$

3.1 Horizontal array

Horizontal array is obtained when all processors of the first column are merged into processor A, processors of the second column into processor B, etc, as presented in Fig. 11. Processors perform the same operations, as before the transformation, beside that, there is an additional input from one of its outputs.

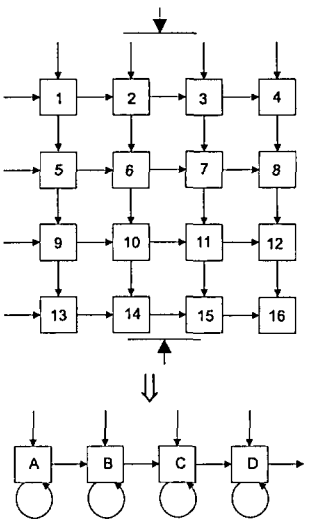


Figure 11: Transformation into horizontal array

Occupation of the processors is presented in Table 6, where numbers represent the processor of the adequate (square) array that would be used in that moment.

Table 6: Processor occupation in horizontal array

	A	B	C	D
1	1			
2	5	2		
3	9	6	3	
4	13	10	7	4
5	1	14	11	8
6	5	2	15	12
7	9	6	3	16
...				
17		14	11	8
18			15	12
19				16

Due to the processor merging the data inputs are changed as presented in Fig. 12.

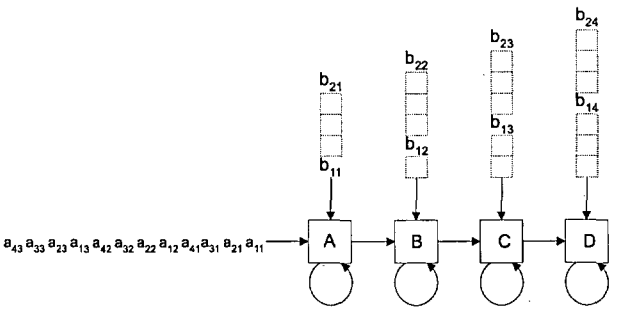


Figure 12: Data inputs in horizontal array

3.2 Vertical array

Vertical array is made when we merge the processors of the first row into processor A, processors of the second

row into processor B, etc, as presented in Fig. 13. Processors perform the same operations as when they were transformed into horizontal array.

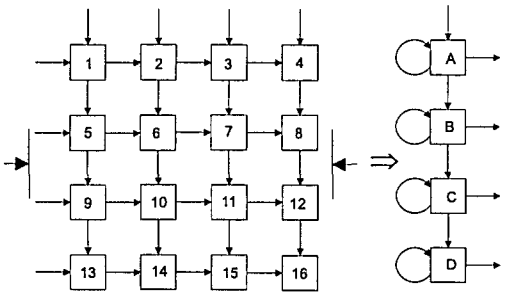


Figure 13: Transformation into vertical array

Occupation of processors is presented in Table 7 and changed data inputs are presented in Fig. 14.

Table 7: Processor occupation in vertical array

	A	B	C	D
1	1			
2	2	5		
3	3	6	9	
4	4	7	10	13
5	1	8	11	14
6	2	5	12	15
7	3	6	9	16
...				
17		8	11	14
18			12	15
19				16

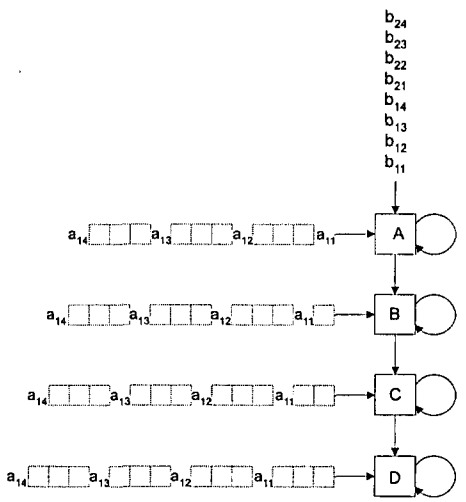


Figure 14: Data inputs in vertical array

Actually there is no significant difference between horizontal and vertical transformation, since all processors in two-dimensional array perform the same operations. Thus, it is insignificant what the contraction direction is, however we can choose which coefficients are delayed when entering the array.

3.3 Diagonal array

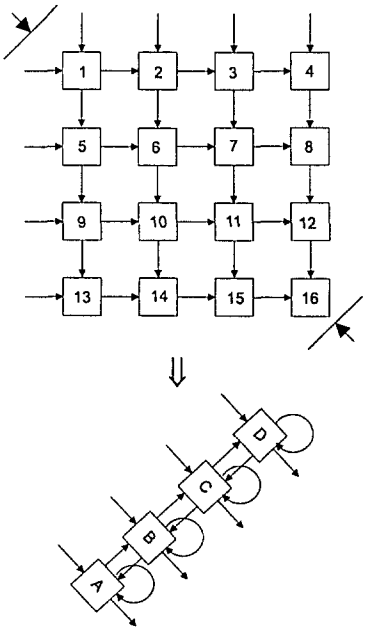


Figure 15: Diagonal transformation (n=4, n^*=4)

Due to the square array structure, diagonal transformation is a bit more complicated. According to the merging process, there can be different linear solutions, but only some typical will be presented in this paper.

If there is an even number of processors (e.g., n=4) in a two-dimensional array, we can choose between two possibilities.

In the first one, as presented in Fig. 15, the processor array is transformed as follows: processors 9, 13 and 14 are merged into processor A, processors 1, 5, 10, 11 and 15 are merged into processor B, processors 2, 6, 7, 12 and 16 are merged into processor C and processors 3, 4 and 8 are merged into processor D. So there is even number of processor (n^*=4) in linear processor array.

Table 8 represents the occupation of the processors, while data inputs are set as presented in Fig. 16.

In the second case, there is an odd number of processors (e.g., n^*=5) in the linear array. According to Fig. 15, processors are merged as follows: processors 9, 13 and 14 are merged into processor A, processors 5,10 and 15 are merged into processor B, processors 1, 6, 11 and 16 are merged into processor C, processors 2, 7 and 12 are merged into processor D and processors 3, 4 and 8 are merged into processor E.

Processor occupation is shown in Table 9, while Fig. 17 presents the data inputs.

But when there is an odd number of processors (n=5) in the two-dimensional array, the linear array consists of odd number of processors(n^*=5) . The situation is presented in Table 10.

Here processors 11, 16, 21, 22 and 23 are merged into processor A, processors 6, 12, 17, 18 and 24 into processor

Table 8: Processor occupation (n=4, n^*=4)

	A	B	C	D
1		1		
2		5	2	
3	9	1	6	3
4	13	5	2	4
5	9	1	6	3
6		10	2	
7	14	5	7	
8	13	1	6	8
9	9	10	2	4
10	14	5	7	3
11	13	11	6	8
12	9	10	12	4
13		15	7	3
14		11	16	
15	14	10	12	8
16	13	15	7	4
17	14	11	16	8
18		15	12	
19		11	16	
20		15	12	
21			16	

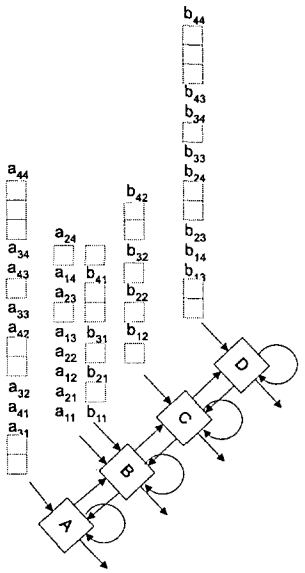


Figure 16: Data inputs (n=4, n^*=4)

Table 9: Processor occupation (n=4, n^*=5)

	A	B	C	D	E
1			1		
2		5	1	2	
3	9	5	6	2	3
4	13	10	1	7	4
5	9	5	6	2	3
6	14	10	1	7	8
7	13	5	11	2	4
8	9		6		3
9	14		11		8
10	13	15	6	12	4
11	9	10	16	7	3
12	14	15	11	12	8
13	13	10	16	7	4
14	14	15	11	12	8
15		15	16	12	
16			16		

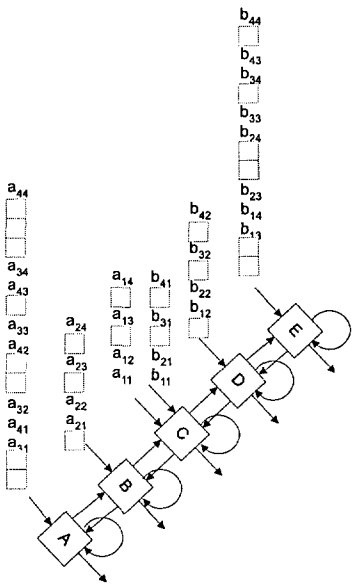


Figure 17: Data inputs (n=4, n^*=5)

B, processors 1, 7, 13, 19 and 25 into processor C, processors 2, 8, 9, 14 and 20 into processor D and processors 3, 4, 5, 10 and 15 into processor E.

Data inputs have to be set according to the new processor utilization, as presented in Fig. 18.

4 Conclusions

According to the results, there are important differences when transforming original triangular array in different directions and with different mirrorings. The difference is in execution time, processor utilization and complexity of processor's operations. Table 11 represents characteristics of $n = 4$ and $n = 5$ arrays. Different transformations are considered (horizontal, vertical, diagonal, interweaved) and different mirrorings (processor A mirrored into processor B, processors A and B mirrored into processor C, ...). Number of steps is the number of systolic cycles needed to perform the algorithm. Number of processors is the number of needed processors, and utilization is their use according to the number of steps (min and max utilization represent smallest and largest utilization of a single processor).

As it can be seen in Table 11 and Fig. 19, mirroring improves the differences between the smallest and largest processor utilization in the array.

Table and figure show these conclusions:

- The number of steps, to execute the algorithm, increases with the transformation, but the number of processors decreases significantly, while their utilization is increased.

- When transforming triangular arrays with even number of processors in the first row of the array, the best transformation is diagonal one with mirroring. Diagonal transformation is the best even if there is no mirroring.

Table 10: Processor occupation (n=5, n^*=5)

	A	B	C	D	E
1			1		
2		6	1	2	
3	11	6	1	2	3
4	11	6	7	2	3
5	16	12	1	8	4
6	11	6	7	2	3
7	16	12	1	8	4
8	21	17	7	9	5
9	11	6	13	2	3
10	16	12	7	8	4
11	21	17	13	9	5
12	22	18		14	10
13	11				3
14	16	12	7	8	4
15	21	17	13	9	5
16	22	18	19	14	10
17	23				15
18	16	12		8	4
19	21	17	13	9	5
20	22	18	19	14	10
21	23	24	13	20	15
22	21	17	19	9	5
23	22	18	25	14	10
24	23	24	19	20	15
25	22	18	25	14	10
26	23	24	19	20	15
27	23	24	25	20	15
28		24	25	20	
29			25		

- When transforming triangular arrays with odd number of processors in the first row of the array, the best transformation is interweaved, while it offers largest utilization and needs only a few processors.
- When transforming square arrays any transformation is better than initial array. Since all processors perform the same operations it is irrelevant in which direction we contract the array, but horizontal or vertical arrays are much simpler to implement than diagonal.
- Processor utilization can be even higher if there are consecutive multiplication computations used one after another.
- In all untransformed arrays the number of steps is defined as $3n-2$ and the number of processors is $\frac{n(n+1)}{2}$ in triangular and n^2 in square arrays, where $n \times n$ is the size of matrix.
- In transformed arrays the number of steps is defined as n^2+n-1 and the number of processors is n.

In some common problems there are very big matrices, e.g., 250 × 250, which lead to the large number of processors required. Therefore in those cases it is appropriate to use also some other techniques with even fewer number of processors [5, 10].

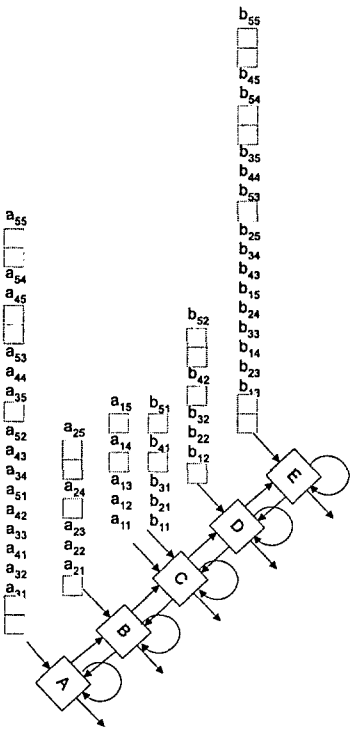


Figure 18: Data inputs (n=5, n^*=5)

References

[1] H.Barada, A.El-Amawy, Systolic architecture for matrix triangularisation with partial pivoting, *IEEE Proc., Vol. 135, Pt. E*, No. 4, July 1988, pp. 208-213.

[2] P.Blaznik, J.Tasič, D.J.Evans, Parallel Solving the Updated Linear Systems of Equations, *F.Solina and B.Zajc (ed.), Proceedings of the second Electrotechnical and Computer Science ERK'93*, Volume B, 1993, pp. 115-118.

[3] J.Kaniewski, O.Maslennikow, R.Wyrzykowski, VLSI implementation of linear algebraic operations based on the orthogonal Fadeev algorithm, *Parallel Computing: State-of-the-Art and Perspectives*, Elsevier, 1996, pp. 641-644.

[4] S.Y.Kung, *VLSI Array Processors*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[5] J.G.Nash, S.Hansen, Modified Faddeeva Algorithm for Concurrent Execution of Linear Algebraic Operations, *Proc. IEEE Transactions on Computers*, Vol. 37, No. 2, February 1988, pp. 129-136.

[6] J.G.Nash, C.Petrozolin, VLSI Implementation of a Linear Systolic Array, *Proc. 1985 Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, pp. 1392-1395.

[7] N.Petkov, *Systolic Parallel Processing*, North-Holland, Amsterdam, 1993.

[8] P.Quinton, Y.Robert, *Systolic Algorithms & Architectures*, Prentice-Hall, UK, 1989.

[9] R.Wyrzykowski, Processor arrays for matrix triangularisation with partial pivoting, *IEEE Proc.-E*, Vol. 139, No. 2, March 1992, pp. 165-169.

Table 11: Array efficiency

	number of		processor utilization (%)		
			overall	single	
	steps	procs.		min	max
n=4					
triangular	10	10	40.0	40.0	40.0
- horizontal	19	4	52.6	21.1	84.2
A into B	19	3	70.2	63.2	84.2
A into D, B into C	27	2	74.1	74.1	74.1
- vertical	19	4	52.6	21.1	84.2
D into C	19	3	70.2	63.2	84.2
D into A, C into B	27	2	74.1	74.1	74.1
- diagonal	16	4	62.5	25.0	100.0
D into A, C into B	24	2	83.3	83.3	83.3
square	10	16	40.0	40.0	40.0
- horizontal	19	4	84.2	84.2	84.2
- vertical	19	4	84.2	84.2	84.2
- diagonal (n*=4)	21	4	76.2	57.1	95.2
- diagonal (n*=5)	16	5	80.0	75.0	100.0
n=5					
triangular	13	15	38.5	38.5	38.5
- horizontal	29	5	51.7	17.2	86.2
A into B	29	4	64.7	51.7	86.2
A and B into C	34	3	73.5	58.8	88.2
- vertical	29	5	51.7	17.2	86.2
E into D	29	4	64.7	51.7	86.2
E and D into C	34	3	73.5	58.8	88.2
- diagonal	29	5	51.7	17.2	86.2
D into C, E into B	41	3	60.9	60.9	60.9
- interweaved	29	3	86.2	86.2	86.2
square	13	25	38.5	38.5	38.5
- horizontal	29	5	86.2	86.2	86.2
- vertical	29	5	86.2	86.2	86.2
- diagonal (n*=5)	29	5	86.2	86.2	86.2

[10] R.Wyrzykowski, Y.Kanevski, S.Ovramenko, Dependence graph transformations in the design of processor arrays for matrix multiplications, *Microproces. & Microprogram.*, Vol. 135, 1992, pp. 534-539.

[11] R.Wyrzykowski, J.S.Kanevski, H.Piech, One-dimensional processor arrays for linear algebraic problems, *Proc. Comput. Digit. Tech.*, Vol. 142, No. 1, January 1995, pp. 1-4.

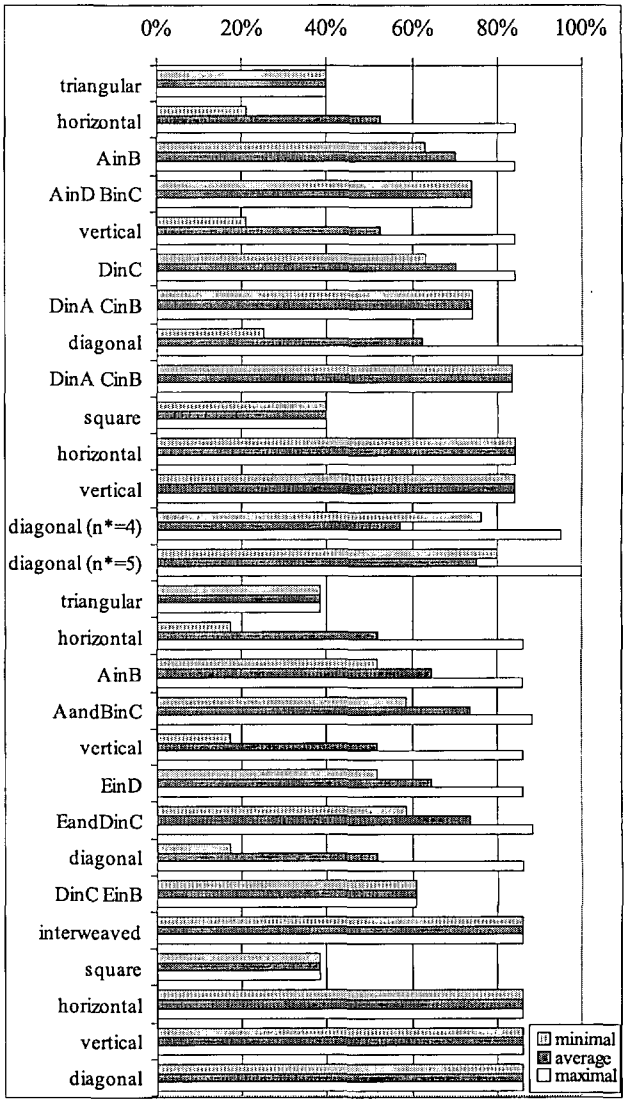


Figure 19: Arrays efficiency

Performance Evaluation of a Hybrid ATM Switch Architecture by Parallel Discrete Event Simulation

Csaba Lukovszki, Róbert Szabó and Tamás Henk
High Speed Networks Laboratory, Department of Telecommunication and Telematics
Technical University of Budapest, Pázmány P. sétány 1/D, Budapest, H-1117 Hungary
Phone: +36 1 463 2764, Fax: +36 1 463 3107
E-mail: {lcsaba, szabor, henk}@ttt-atm.ttt.bme.hu

Keywords: ATM switching, QoS, performance evaluation, parallel modeling

Edited by: Matjaž Gams
Received: August 31, 1999 **Revised:** August 31, 1999 **Accepted:** August 31, 1999

Nowadays, telecommunication networks are passing through a rapid evolution. The introduction of novel services in high-speed networks such as Asynchronous Transfer Mode (ATM) requires different handling of connections. While real-time applications usually require low end-to-end delay and delay jitters, and are not very sensitive to loss within an acceptable range, there exist applications that require very low loss rates but are not concerned of delay or delay jitters. The concept of designing Multistage Interconnection Network (MIN) capable of operating with and without cell loss and cell delay sensitiveness can be a reasonable alternative.
This paper is divided into two subjects. In the first a new switch architecture is depicted. Our proposed switching architecture and protocol handles connections' different Quality of Service (QoS) requirements with different mechanisms. We introduce four cell types that can be mapped to the existing ATM services. The investigation of such a complicated system as an ATM switch, requires lot of simulation time in any conservative simulation environment. To achieve shorter simulation time, our simulation platform was designed for Parallel Discrete Event Simulation (PDES) environment. The extended features and mechanisms of our MIN involves individual solutions during interaction among processes, which are addressed in the second part of the article.

1 Introduction

Communication on Broadband-Integrated Service Digital Networks (B-ISDN), such as Asynchronous Transfer Mode (ATM), demands the network to handle different connections differently. In accordance with expectations of endpoint user, connections of B-ISDN covers voice, data, video and image traffic within a single network. The introduction of these novel services into high-speed networks such as ATM, requires different handling of connections [5, 12, 17]. The two significant classes of connections are established by real-time and data applications. The main difference between these two is based on the cell loss and delay sensitiveness. While real-time like applications usually tolerate cell loss in order to achieve low delay and delay variation, on the other hand data applications require very low loss rate with indifference in delay and delay jitters.
The optimization of a switching architecture based upon the above mentioned two criteria requires different demands. Low delay and delay variation can only be achieved with small buffer sizes in the network, but it results in higher cell loss in case of bursts [17, 18]. As a trade-off between these requirements, switches are usually optimized for one of these criteria, and some other mechanism (e.g. priority scheduling) is used to ensure the other criteria.
Multistage Interconnection Networks (MINs) offer a reasonable alternative between cost and performance. Most recently, there are growing interests in using MINs as the interconnection structure of the switching nodes of high-speed communication networks [11, 12, 13]. MIN is an example of Banyan networks - see Figure 1 -, where sufficiently large buffers are applied at the outputs of the Switching Elements (SEs), and a priority scheduler is used to ensure low delay for delay sensitive traffic. However, in order to meet the very low cell loss ratio requirements buffer spaces need to be increased, which introduces both delay and delay jitters. This architecture has the disadvantage of the necessity of large buffers in each SE.
The counter example can be a backpressure MIN that uses an additional, two directional communication protocol in order to stop forwarding cells when there is not enough free buffering capacity in the next-stage SE. In this way this backpressure mechanism ensures that there will be no cell losses inside the switch. Besides, one must face the problem of maximum achievable throughput, i.e: simple backpressure MINs have a theoretical maximum in throughput around 60% of link capacity [13, 17].
Our proposed architecture mixes two protocols in the same switching fabric in order to be optimized for the differ-

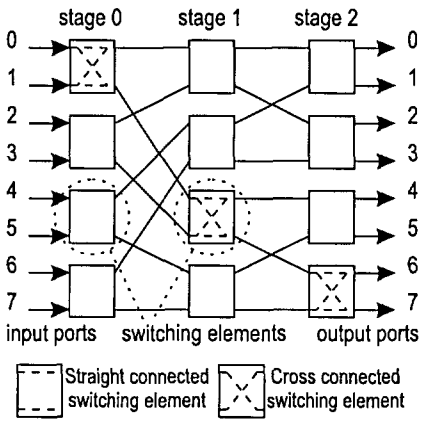


Figure 1: Multistage Interconnection Network (MIN)

ent type of services. For data applications, a backpressure mechanism is used to reduce cell loss ratio. For real-time applications, higher priority class is applied with small buffers and no backpressure mechanism.

The proposed switching architecture and protocol is modeled in a parallel simulation environment, where exaggerated precautions are necessary because of the mixed protocol scheme in order to avoid dead-locks. The rest of this article deals with the performance evaluation of the proposed hybrid switch architecture. First, we describe the model of our hybrid MIN and identify the requirements put on it. Then the simulation environment that has been developed for performance analysis is described. Finally, simulation results are discussed and conclusions are drawn.

2 The Model of the Hybrid MIN

We propose a novel switching architecture that can distinguish among cells from different kinds of connections in cell level. To achieve our goal some modifications was accomplished inside the switching elements. In this way the proposed MIN is capable of operating with loss and delay sensitive cell flows simultaneously. Thereby, hybrid MIN can make difference among connections inside the switching fabric and efficiently utilizes the performance of the switch.

The concepts for designing a MIN capable of operating with internal cell loss and with backpressure mechanism comes from the different Quality of Service (QoS) requirements imposed on ATM. There are six QoS parameters defined by the ATM Forum that identifies the end-to-end behavior of a connection [1]. As three of these are independent of the switch architecture, we only consider Cell Delay Variation (CDV), Cell Transfer Delay (CTD) and Cell Loss Ratio (CLR) (see Table 1).

Accordingly, to investigate MINs it was necessary to map the connection level QoS parameters into a cell level equivalent. We distinguished four types of cells inside our switching fabric (see Table 2).

The mapping was chosen in order to optimize the favorable

	CBR	rtVBR	nrtVBR	ABR	UBR
CDV	x	x			
CTD	x	x			
CLR	x	x	x	x	

Table 1: QoS parameters

	Loss Sensitive	Delay Sensitive
type A	x	x
type B	x	
type C		x
type D		

Table 2: Distinguished cell types

parameters of each individual MIN type. Loss sensitiveness can be handled by the backpressure protocol, while low delay and delay jitters can be guaranteed with the priority mechanism. Besides, backpressure mechanism introduces extra delays that can be eliminated by applying higher priority class as well.

Further, during the design of the hybrid MIN some necessary architectural constraints were to be fulfilled, i.e. backpressure MIN architecture requires input port buffering to support up-to-date information on the available resources [13, 14]. However, MIN with traditional input port buffering has significant disadvantage in throughput caused by head-of-line blocking [13, 14, 20]. Hence, our switching elements (see Figure 2) consist of both input and output buffers.

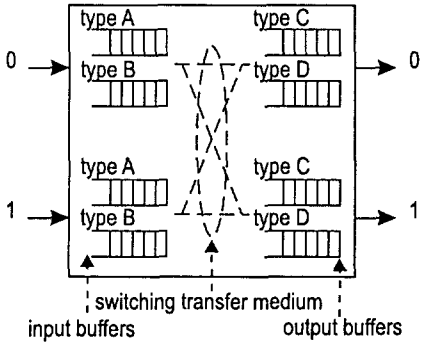


Figure 2: Switching Element (SE)

According to the Figure 2, each cell type has a dedicated buffer through every path inside the SE. The two loss sensitive traffic types (A, B) have buffers at the input ports and are handled with the backpressure protocol, while the delay sensitive cells of type C and D are just forwarded and stored at the output queues or dropped in case of congestion.

Besides, at each output port we use scheduling with static priority scheme [8]. The service order is C, A, D, B. It means whenever there is at least one cell in buffer associated with type C, it is served first, otherwise cell is chosen according to the priority order. It stands to reason, culling a cell from some input buffer is possible if that cell is des-

tined to the output port where scheduling is being executed. Besides, in order to increase efficiency at higher loads, if the forwarding of a type A cell was blocked due the backpressure mechanism, the server checks for a lower priority cell to transmit. On the whole, the forwarding of the C cell type is straightforward.

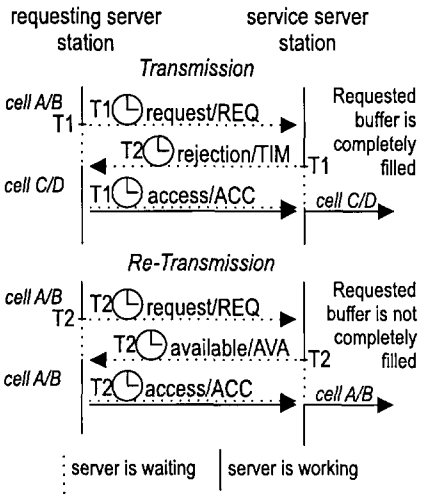


Figure 3: Backpressure protocol

Figure 3 describes the forwarding of a backpressure cell A or B. In the first case, the requested buffer type is completely filled, so an access rejection message is sent back to the previous stage. Now, if we have a lower priority cell D, or a type C cell has arrived during the blocked period, that is forwarded instead of the blocked cell. In the second case if the request is confirmed with an available message, the original cell is forwarded.

3 The Parallel Model of Hybrid MIN

Uniprocessor systems are approaching their limit on computation speed. The computation power of such systems can not be improved infinitely due to the fact that the speed of a signal transmission on a chip is bounded by the speed of light [16]. In the last two decades the interests in multiprocessor systems consisting of hundreds, or even thousands of processors has increased noticeably. The interaction among these processors is realized either by message passing or by shared memory [7, 10].

For this reason, it seemed a promising directive to reduce the MIN's simulation runtime by exploiting parallelism in the switching fabric and mapping the simulation program onto multiprocessor machines. Moreover, the architecture of MINs with their identical switching elements also lends themselves for parallel computation. Additionally, parallel simulation is also interesting from an academic point of view, because it represents a problem domain that often contains substantial amounts of parallelism, yet paradoxically, it is surprisingly difficult to parallelize in practice.

In the case of parallel simulation one can follow either the optimistic or the conservative approach [14]. We chose the conservative approach, where no causality errors are allowed at all because of its less complexity. Here one must ensure that the execution of events is in strictly non-decreasing time-stamp order. However, to ensure this non-decreasing time-stamp order execution of events in concurrently executed threads, processes have to be blocked until it is safe to execute the next event in their event lists. Safe means there is no chance to get an event with time-stamp earlier than the executed one. However, the blocking and invoking of processes can result in circular wait deadlock situations that have to be avoided by good design. On the other hand, in the optimistic approach where no precautions are taken to avoid the causality error one has to detect and correct it somehow. Usually a roll back mechanism is used to enforce the in-order execution of the critical events.

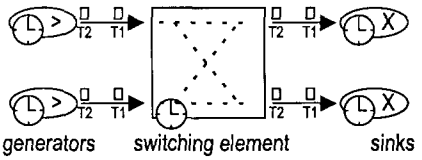


Figure 4: Process oriented simulation model

The simulator, that we designed is a shared memory model conservative parallel simulator, where each switching element was mapped to a thread. Consequently all switching elements have their own local clock that indicates their progress in simulated time, state variables that describe the state of the system and an event list containing all pending events for the specific SE, that have been scheduled but have not taken effect yet. The basic component, i.e. a 2x2 switching element connected to traffic generators and sinks can be seen in Figure 4. Here the 5 physical objects were mapped to 5 logical threads of the simulation.

The simulation model for the Banyan switch [11], where no backpressure mechanism is used, is quite straightforward. In this case, messages are only passed in the forward direction. The very special MIN architecture with this unidirectional message passing property requires minimal synchronization among the processes and is very suitable for parallel computation [9, 13, 15, 16]. However, as one can not apply infinite message storing capacity among the processes, it is necessary to support a minimal synchronization among the processes, unfortunately introducing dead-lock into this very simple scheme, too [6, 13].

The backpressure MIN itself requires very strict synchronization among the stages of the MIN, which results in a synchronization at the level of threads also. In order to ensure no internal loss within the switching fabric backpressure cell forwarding must adhere to the following protocol:

1. Before a cell transmission is initiated, the sender must find out if the receiving buffer contains at least one empty slot. This is done by sending a request (REQ)

message, and waiting for the reply.

2. When a SE receives a REQ message, it checks for empty slots in the corresponding buffer of its input ports. If the buffer is not completely filled, an available (AVA) message is sent back to the sender. Otherwise a time indication (TIM) signal is sent with the indicator of the earliest time when the desired buffer could be released. The switching element can proceed with the next event of its event list, but must check back to the refused port at the time indicated by the TIM message for further REQ message.

3. The switching element must wait for an access (ACC) message after it replied an AVA message.

4. A switching element can only send out an ACC message if an AVA message has been previously received.

5. A switching element receiving TIM message can proceed with the next event of its event list, but must schedule the re-transmission of the blocked cell at the time indicated in TIM message.

An illustration of the proposed backpressure transmission protocol can be seen in Figure 3.
Because of the almost continuous synchronization among the processes the achievable speed-up in backpressure MIN simulation is lower than in case of traditional Banyan MIN [11, 13]. In our special case, we wanted the synchronization to be as loose as possible, so we synchronized the processes only when they were playing the backpressure handshaking protocol.
Our backpressure mechanism extends the above detailed protocol by utilizing the blocked period. Whenever a requesting server receives a TIM message it checks the buffers for other cell types. If during the blocked period there is at least one cell in other buffers, the scheduling mechanism chooses one cell to forward according to the priority order. Necessarily, the backpressured cell has to be scheduled to re-transmission after the chosen cell leaves the SE.

4 Simulator Model of the Switch Architecture

The simplest investigated architecture consists of a MIN switching fabric - interconnected switching elements -, cell generators and sinks that are connected to the inlets and the outlets of the MIN, respectively. The mapping of the model onto the parallel simulation is made by the assignment of logical processes to each individual building blocks i.e. SEs, generators and sinks. As described earlier, logical processes have their local clock, event list and state variables. Cells are passed through the architecture by modeling *DEPARTURE* and *ARRIVAL* events of the corresponding cell types.

4.1 Sink and Generator Model

Cell generators are used in the simulation as traffic sources. They can produce cell streams different ways in conformity with the modeled traffic source. They can substitute some often used arrival types, like Poisson Processes (PP), Interrupted and Markov Modulated Poisson Processes (IPP, MMPP) and constant arrivals. Constant arrival model is used to model Constant Bit Rate (CBR) traffic, Variable Bit Rate (VBR) traffic is modeled by PP, IPP or MMPP depending on its characteristics. A generator connected to an inlet of the MIN can produce all our four cell types acting like a multiplexer at the same time. With this we reduce the number of concurrently running processes, thus also reducing the complexity. Sinks are only used in our model to gather performance statistics on MIN's performance. Besides, they also have to participate in transmission protocol with the last stage SEs.

4.2 Switching Element Model

As we already mentioned, switching elements must have an event list. However, we decided to distribute our event list to the corresponding ports. These event lists are shared objects as they are manipulated by two simulation objects i.e. the owner and the one the port connects to. As processes have to process events in non-decreasing time-stamp order, ports have to be sorted to offer the lowest time-stamp event. If any of the input ports are empty, process must be halted until a new event arrives at the corresponding port to avoid causality error. The processing of the selected event depends on its type. *DEPARTURE* events initiate cell forwarding either with backpressure or traditional mechanism. After a successful cell transmission a new already buffered cell has to be scheduled for transmission. Here, buffers are selected according to their priority order see the previous section. *ARRIVAL* event must accept or reject the forwarding of a cell depending on the used protocol. After receiving a cell, it must be queued to the corresponding buffer.

5 The Simulation Platform

The simulator itself was developed in a C++ extension called micro (μ) C++ [3, 4]. This extension introduce new object types that augment the exciting panoply of control flow facilities and provide for lightweight concurrency on single processor and parallel execution on multiprocessor computers running the UNIX based operating systems. The μC++ programming language answers the requirements of parallel programming by providing wide range of possibilities of interaction among different kinds of objects. This allows the programmer to choose the kind of object best suited to the particular problem without having to cope with communication restrictions. Besides, this programming language facilitates synchronous and asynchronous communications not hiding their details in a system, which increases the flexibility.

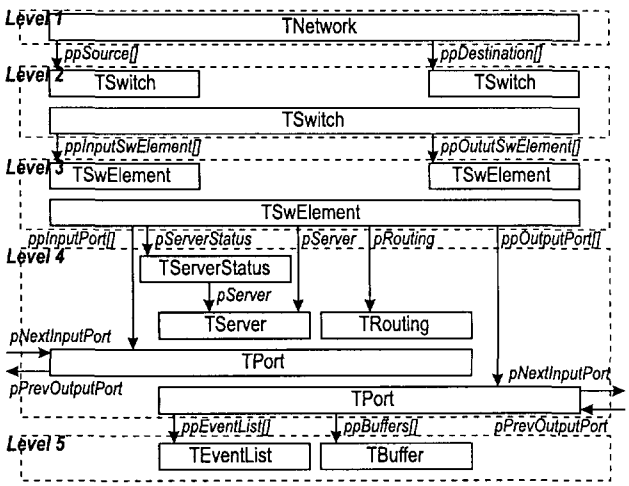


Figure 5: Object model

The object structure was designed to support simulation of different MIN based ATM switching fabrics with backpressure or traditional forwarding protocols. Both the structure and objects' functions are flexible enough to conform to these requirements. Each object realizes an organic part of duties in the switch. The basic building blocks and their connections can be seen in Figure 5. The figure also represents the hierarchical structure of objects and how a SE and a MIN is constructed. Some objects implement only PDES functionality and others realize switching functions. As an example the *TPort* object plays an active part of the switching providing synchronous and asynchronous transmissions of cells. The simulation platform was designed to be a flexible and scalable simulation platform supporting simulation of different ATM switch architectures. It enables the user to easily configure the network topology and to implement new functions by deriving objects from the existing base classes.

Basically the *TSwElement* object represents a SE. The underlying objects implements specific features for SE and the parallel execution as well. The *TRouting* object is responsible to route cells to the destined outlet. The *TPort* component is designed to keep connection with other SEs, otherwise it plays the backpressure protocol with other SE's ports. The objects at Level 5 are destined to store cells and events respectively. The levels above the *TSwElement* facilitate to simulate big interconnected switching networks.

A Network Configuration File (NCF) was designed to automate topology configurations. By editing this input file, one can create a top-down network topology without any knowledge of underlying PDES mechanism. On the other hand, experts can easily extend the existing library functions and basic classes to derive new, enhanced features.

6 Traffic Models

Packet generators are used in the simulation as traffic sources. They can produce cell streams different ways in conformity with the modeled traffic source. They can act as if they were some CBR applications producing cell stream with constant inter-arrival time. To model applications characterized with VBR, generators can produce cell streams according to PP with arrival rate λ. On the other hand to model real-time like application, generators bear the ability to produce cell streams according to a MMPP with average sojourn times ν_1^{-1} and ν_2^{-1}, for states 1 and 2, respectively. When the Markov chain of a real-time application is in state i ($i = 1, 2$), the arrival process from that source is Poisson with rate λ_i. The stationary probability vector of the Markov chain is:

$$\bar{\varepsilon} = [\varepsilon_1, \varepsilon_2] = [\frac{\nu_2}{\nu_1 + \nu_2}, \frac{\nu_1}{\nu_1 + \nu_2}] \quad (1)$$

where ε_i is the stationary probability in state i.
The effective arrival rate from one such a source is:

$$\lambda_{eff} = \sum_{i=1}^{2} \varepsilon_i \times \lambda_i \quad (2)$$

In our first experiment we only tested our hybrid MIN model against results found in literature. Here we used Poisson generators for all four cell types. However, in our next experiment in order to be consistent with the introduced four cell types (A, B, C and D) we used different traffic sources in accordance with the modeled traffic type. For type C cells constant CBR generators were used; for type A traffic IPP were used to emulate rtVBR cell streams; for type D cells (rtVBR and nrtVBR) MMPP generators were used while simple Poisson sources were used for cell type B.

7 Numerical Results

7.1 Experiment 1

In the first Experiment we only testified our simulator against the results found in related literature [2, 19]. We present results for a 8x8 MIN made of 2x2 SEs. We investigated maximum achievable throughput, delay and delay jitters of the different cell types. Besides, we present some results on how the buffer allocation influences these characteristics. The throughput of the switch architecture is given by:

$$\text{throughput} = \frac{\text{\#of arrived cells}}{\frac{\text{simulation time}}{\text{service time}} \text{\#of ports}} \quad (3)$$

where *#of arrived cells* is the summed number of arrived cells at the outputs, *simulation time* is the duration of simulation, *service time* is the service time of a cell and *#of ports* is the number of output ports.

As for the input traffic, we connected 4 generators - one for each cell type - through a multiplexer to each input port of the MIN. They generated cell streams according to Poisson arrivals with identical rate. The aggregated arrival rate of the cells are shown in the figures as generator intensity. The destinations of the cells were also identically distributed.
On Figure 6 we show how our hybrid MIN's throughput goes compared to an ordinary backpressure MIN. During the simulation we applied 4 buffer slots at each input ports of the backpressure MIN and 2 buffer slots for each cell type inside our hybrid MIN.

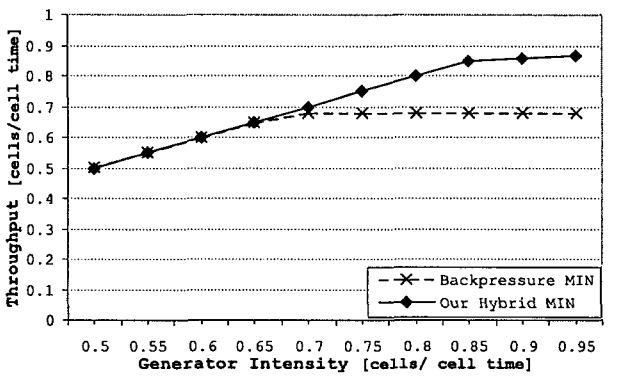


Figure 6: Throughput comparison between Backpressure MIN and our proposed hybrid MIN

It is quite obvious from the figure, that our hybrid MIN's throughput characteristic is better than that of the backpressure. However, we must admit, that it is only true if not the type A and B traffics are dominant. In the most extreme case, when we only have the type A and B cells, we would obtain the same throughput as a backpressure MIN. Fortunately, this scenario is not typical at all.

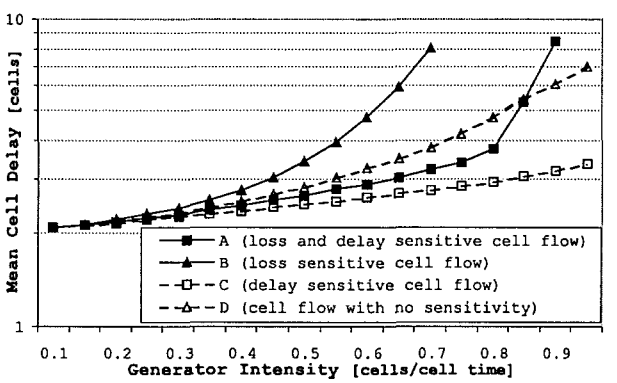


Figure 7: Mean Cell Delay for cell types, while buffer size for each cell type is 10

In Figure 7- 8, one can see how our static scheduling works. The lowest priority traffic, i.e. the type B cells, is punished most according to Figure 7, but still here we guarantee no cell loss inside the switching fabric. It can be efficient or

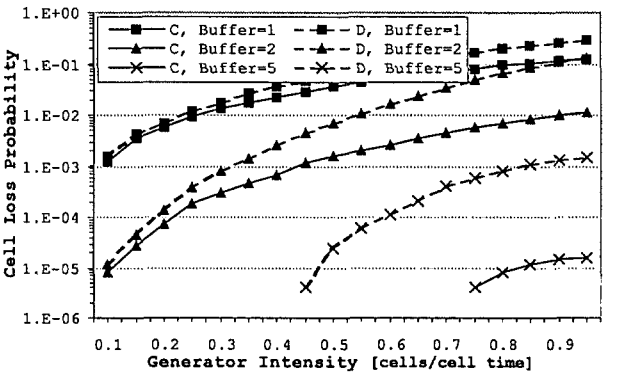


Figure 8: Cell Loss Probability

can reduce the expenses of memories to concentrate big buffers only at the ingress ports of the MIN, where the congested cells are backpressed to.
In Figure 8, we show the cell loss probability of type C and D cells. It can be seen that even a buffer capacity of 5 cells for the type C traffic can significantly reduce its loss probability. On the other hand, Figure 8 also shows that the type D buffers should be increased in order to achieve lower cell loss ratio.

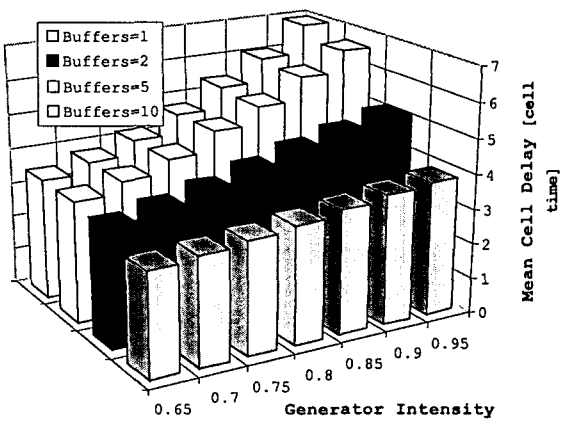


Figure 9: Mean Cell Delay for cell type D

In Figure 9 we present how changing of the applied buffer size for all cell type affects the characteristics of Mean Cell Delay for cell type D. As it was excepted the delay increases as the function of applied buffer size.

7.2 Experiment 2

Further we investigated, how a finite buffer capacity should be partitioned among the different cell types to achieve the best performance. Here, we also took into account, beside our priority scheme, that we want to minimize the delay and delay jitters of the type C and A traffic classes. So we kept in mind, that for cell types A and C buffers should be smaller than for the types B and D. One of our partition-

ing scheme is shown in Table 3, where we examined four buffer partitioning realizations with a capacity of 60 cells (Table 3: set 1-4).

buffer size	set 1	set 2	set 3	set 4
type A	1	2	3	4
type B	4	9	14	19
type C	11	10	9	8
type D	44	39	34	29

Table 3: Buffer allocation schemes

From the results shown in Figure 10 and 11, it was interesting to see, that even though the type A cell is ahead of the type D cell in the service priority order, it suffered higher delays if the backpressure mechanism was in action for the most part of the operation (see Figure 10). This can be explained by the fact, that if a type A cell was stopped forwarding because of the lack of buffer space at the next stage, a lower priority type D cell was transmitted if it was queued in the system.

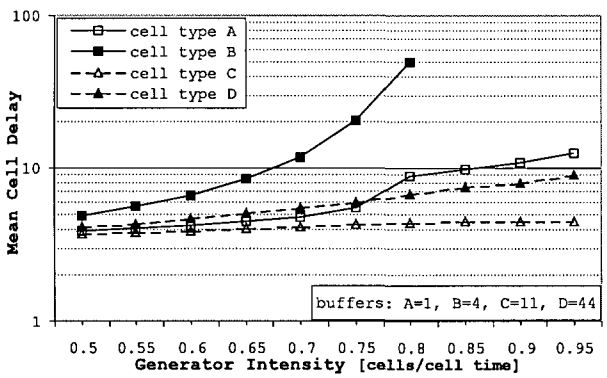


Figure 10: Mean Cell Delay for Set 1

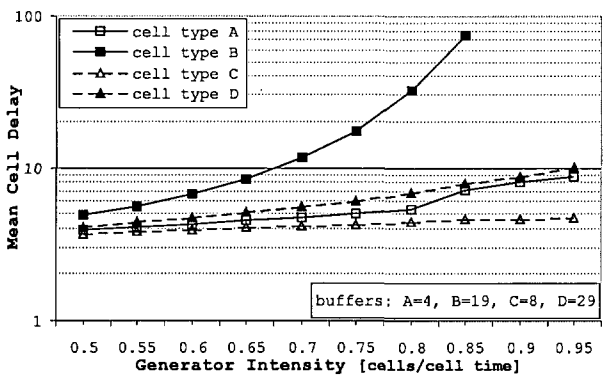


Figure 11: Mean Cell Delay for set 4

In Set 4, where we allocated more buffers to type A cells, their delay went just below the delay of type D cells (Figure 11). The delay of the type C cells, which were the highest priority traffic, were indifferent to this allocations. We found that with properly adjusted buffers, one can optimize the network performance. One of our recent interest is to investigate, how this optimal allocation should be implemented.

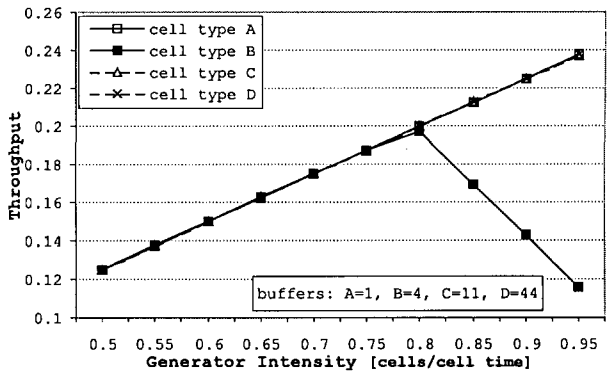


Figure 12: Throughputs for set 1-4

Besides the delay characteristics depicted above, we also examined how the throughput of the overall switch was affected in these buffer allocations. First, Figure 12 shows that the throughput curves are linearly increasing functions of the offered load for all cell types but type B, where we experience a breakdown similar to the backpressure MIN's throughput characteristics. However this breakdown point was dependent on the actual buffer allocation (see Figure 13).

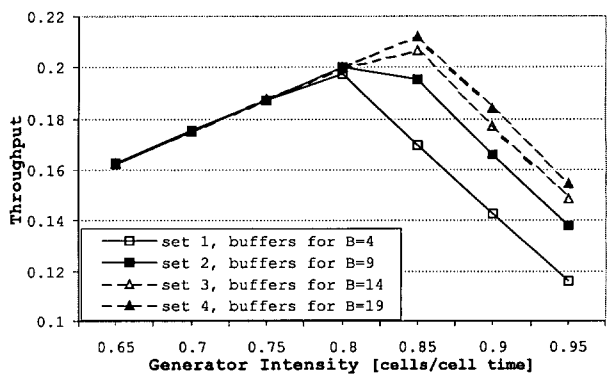


Figure 13: Throughput if cell type B for set 1-4

7.3 Experiment 3

After we testified our simulator and the results in throughput and delay seemed promising we started to analyze traffic situations more realistic than in the previous case. All the following results are shown for 16x16 MINs.
We used traffic models shown in Table 4 to emulate the different characteristics imposed by the diversity of ATM traffics. The total offered load was distributed evenly among the different cell types, so only the total offered load is

presented in charts further on. The load imposed on the switching fabric was also symmetric in the means that all destination ports were equally preferred. First of all we in-

type	Cell stream generators
A	Interrupted Poisson Process–rtVBR
B	Poisson
C	Constant inter-arrival time–CBR
D	Markov Modulated Poisson Process–VBR

Table 4: Streams for cell types

vestigated the buffer requirements of the real time traffic, i.e. type C cells as the highest priority traffic. We found that in order to satisfy the QoS requirements of the real time traffic up to 25% of total load we have to allocate at least 4 or rather 5 cell-size buffer for this class (See Figure 14). The delay of this class remained minimal (5 to 7 cell time) all the time thanks to the static priority applied. After we identified the needs of the highest priority traf-

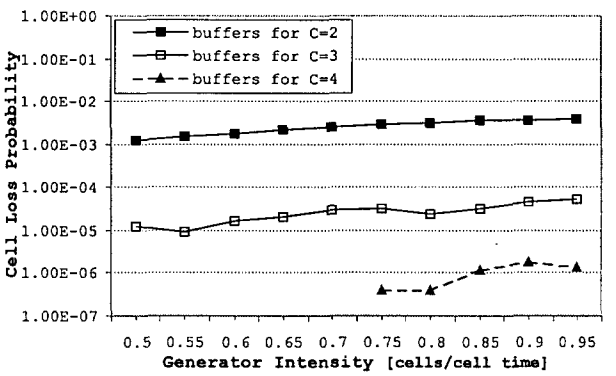


Figure 14: Cell Loss Probability for the CBR traffic class

fic, i.e. type C, we tried to configure optimally the buffer partitioning between the type A and D cell streams (see buffer locations in Figure 2). We further assumed that we can only dispose of a finite and pre-defined buffer space to be partitioned and we tried to find an optimal partitioning by heuristic methods. The correlated results can be seen in Figure 15 and 16. Here the mean delays of A and D cells can be seen respectively for different buffer partitioning. It can be seen from Figure 15 and 16 that the allocation and de-allocation of a few buffer spaces can significantly degrade or improve the performance of the respecting cell type. It is also clear from the results that one must find a trade-off as the improvement of one cell class results in the degradation of the other class. Finding the appropriate buffer partitioning is not the least straightforward. For this reason an adaptive mechanism may be required in the switching fabric to adjust the buffer partitioning to the actual network load in order to achieve optimal performance. The study of this question and to determine a metric to be optimized is our future plan. However, as a primary metric we used the summa mean delays of the first three priority classes i.e. C, A, D to determine a performance descriptor.

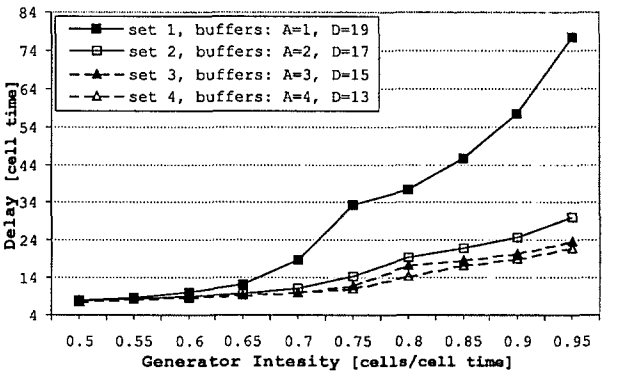


Figure 15: Mean Cell Delay of cell type A

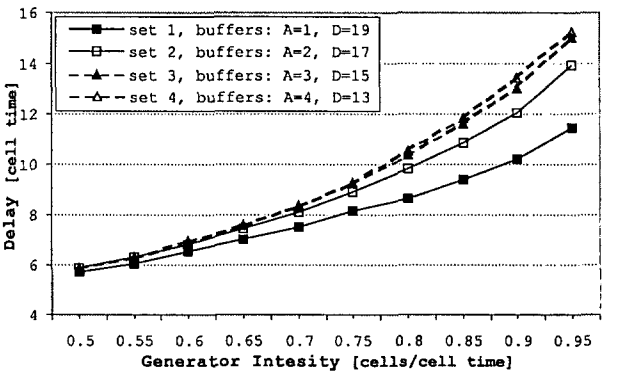


Figure 16: Mean Cell Delay of cell type D

An example can be seen in Figure 17. These charts vary both in the actual offered load and in the realized buffer partitioning scheme. To determine the optimal buffer partitioning for an offered load, one has to compare all possible partitioning settings for the given load to determine the optimal settings. However the changing in load may result in the need of re-partitioning of buffers in order to achieve optimal performance. This comparison is complex and time consuming, so we are seeking for heuristic solutions to maintain optimal buffer partitioning in case of varying loads.

8 Conclusion

We have introduced a new hybrid protocol MIN, which can support the complete range of traffic types existing in ATM networks. We presented some forerunner results as a base of our performance analysis in this field. We found the results good enough for further investigation. We analyzed a 16x16 MIN with different traffic loads to determine the optimal buffer partitioning among different traffic classes. We also proposed a performance metric to optimize the buffer partitioning for. Our future plan is to further investigate the possibilities to determine the optimal buffer partitioning and to simulate large ATM switches.

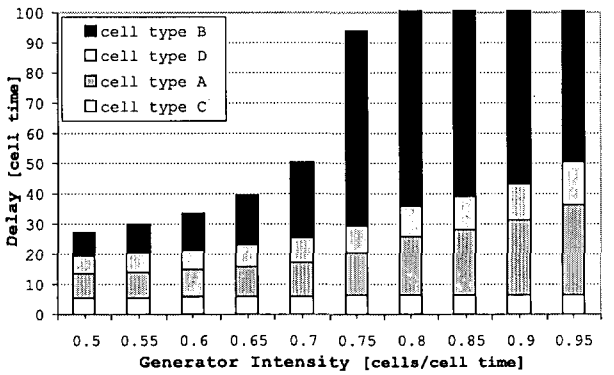


Figure 17: Proposed performance metric

9 Acknowledgement

First of all we want to thank Gábor Fodor, who has planned the idea of a hybrid switch in our mind a few years ago. Besides, this work could not have been done without the support of the High Speed Networks (HSNLab) of the Technical University of Budapest. We also owe our special thanks to József Bíró, for his selfless assistance.

References

[1] ATM Forum (1996) Traffic Management Specification, Version 4.0. *ATM Forum Specification*

[2] Bhagwat S., Tipper D., Balakrishnan K., Mahapatra A. (1994) A Comparative Evaluation of Output Buffer Management Strategies in ATM Networks. *Proceedings of IEEE International Conference on Communications'94*, New Orleans, LA, (May)

[3] Buhr P. A., Ditchfield G., Stroobosscher R. A., Younger B. M.,Zarnke C. R. (1992) C++: Concurrency in the object-oriented language. *C++ Software Practice and experience vol. 22(2)* p. 137-172, (February)

[4] Buhr P.A., Stroobosscher R. A. (1995) μC++ Annotated Reference Manual, Version 3.7., Dept. of Computer Science, University of Waterloo, Ontario, Canada

[5] Denzel J. W. E., Engbersen A. P. J., Iliadis I. (1995) A flexible shared-buffer switch for ATM at Bb/s rates. *Computer Networks and ISDN Systems*, 0169-7552/95

[6] Fodor G. and Szabó R. (1996) Comparison Of Null Message Reduction Techniques In The Parallel Simulation Of Multistage Interconnection Networks *Proceedings of the European Simulation Multiconference '98*

[7] Fujimoto R. M. (1990) Parallel Discrete Event Simulation *Communications of the ACM*, Vol.33, No.10 (October)

[8] Lin A. Y. M., Silvester J. A. (1991) Priority Queuing Strategies and Resource Allocation Protocols for Traffic Control at an ATM Integrated Broadband Switching System *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 9a. p. 1524-1536 (December)

[9] Lukovszki Cs., Szabó R., Henk T., (1998) Parallel Simulation Model of a Novel Hybrid MIN Architecture, *10th European Simulation Conference, ESS'98*, Nottingham, UK, p. 173-179

[10] Nicol D., Fujimoto R. M. (1994) Parallel Simulation Today *Annals of Operations Research*, (December)

[11] Oktug S.F, Caglayan M.U. (1997) Design and Performance Evalution of a Banyan Network Based Interconnection Structure for ATM Switches *IEEE Journal on selected areas in Communications*, Vol. 15. No. 5.

[12] Rooholamini J. R., Cherkassky V., Graver M. (1994) Finding the Right ATM Switch for the Market *0018-9162/94 IEEE COMPUTER*, (April)

[13] Szabó R. (1996) Simulating ATM switches with parallel discrete event simulation *Diploma Work, Technical University of Budapest*

[14] Szabó R., Fodor G. (1996) Parallel Simulation of MINs Without Internal Cell Loss *Hungarian - Swedish - Networkshop '96 - Spring; International Workshop on ATM Networks*

[15] Szabó R., Lukovszki Cs. (1998) Performance Evaluation of a Hybrid ATM Switch Architecture by Parallel Simulation *Conference Paper, SPECT'98*, Reno, USA, (July)

[16] Tay S. C., Teo Y. M. (1994) Conservative Parallel Simulation of Finite Buffered Multistage Interconnection Networks *Technical Report TRE6/94, Department of Information Systems and Computer Science*, National University of Singapore, pp. 24,

[17] Tobagi F.A. (1990) Fast Packet Switch Architectures For Broadband Integrated Services Digital Networks *IEEE Proceedings*, (January)

[18] Yegani P., Krunz M., Hughes H. (1994) Congestion Control Schemes in Prioritized ATM Networks *Proceedings of the IEEE ICC '94 Conference*, New Orleans, p. 1169-1173, (May)

[19] Mun Y., Youn H. Y. (1994) Performance Analysis of Finite Buffered Multistage Interconnection Networks *IEEE Trans. on Computers*, p. 153-162, (February)

[20] Obaidat M. S., Riehl M. (1998) An Evaluation Simulation Study of Input Buffering of ATM Switches *Symposium on Performance Evaluation of Computer and Telecommunication System, SPECT'98*, Reno, USA, p. 58-66. (July)

Overview of Consciousness Research

Imants Barušs
King's College, 266 Epworth Avenue, London, ON, Canada N6A 2M3
baruss@uwo.ca

Keywords: consciouness, academia, science, experience, reality, beliefs, materialism, transcendentalism, anomalies, education

Edited by: Anton P. Železnikar
Received: January 19, 1999 **Revised:** October 5, 1999 **Accepted:** November 30, 1999

The purpose of this paper is to orient the reader to the contemporary scientific study of consciousness. One of the most noticeable features of research concerning consciousness is that there are three domains of discourse, the physiological, computational and experiential, each with its own methodology and concerns. While confusion is often expressed about what it is that one is discussing, there are four main categories of definitions of the term consciousness: consciousness$_1$ is the registration, processing and acting on information; behavioural consciousness is the explicit knowledge of one's situation, mental states or actions as demonstrated by one's behaviour; subjective consciousness is the subjective stream of thoughts, feelings and sensations that occur for a person; and consciousness is the sense of existence of the subject of mental acts. There are also disparate views concerning consciousness that surveys have revealed to be correlated with investigators' beliefs about the nature of reality along a material-transcendent dimension. Those with materialist views tend to think that only that which is physical is real and that consciousness is an emergent property of neural or information-processing systems; those with conservatively transcendent views think that there is more to reality than that which is physical and emphasize subjective aspects of consciousness; while the extraordinarily transcendent conceptualize consciousness as ontologically primitive and place importance on self-transformation. An investigator's contention that she has had anomalous experiences appears to incline her toward a transcendent position. The presence of these correlations indicates that research programs concerning consciousness proceed, not in an unbiased manner, but on the basis of personal beliefs about the nature of reality. Can beliefs change in the course of the educational process? Data from 129 undergraduate students indicates that beliefs about consciousness and reality can move in a transcendent direction in classes with an instructor with extraordinarily transcendent beliefs.

1 Introduction

There has recently been a great deal of interest concerning consciousness within the academic community, yet the research effort has been fragmented with many academics working at cross-purposes to one another. What is presented here is an overview of the contemporary study of consciousness that can serve to provide a context for discussions concerning consciousness. This consists of a delineation of three domains of discourse, metanalysis of definitions of consciousness, a discussion of the beliefs about consciousness and reality of consciousness researchers and the results of a study concerning changes in students' beliefs that has implications for consciousness research.

2 Domains of Discourse

One of the most noticeable features of research concerning consciousness is that there are three domains of discourse that often have little to do with one another. One domain of discourse, the physiological, is concerned with an understanding of the biological processes involved in consciousness. This is the realm of neuroscience and the usual methods of biology and observation of behaviour are used in order to gather knowledge concerning consciousness. Somewhat disconnected from neuroscience, although often considered part of the physiological domain, are discussions of the relationship between subatomic events and consciousness. This includes discussions of relationships between quantum mechanics and mind.

A second domain of discourse is the computational whereby consciousness is discussed in terms of information processing. This is the area of cognitive science which subsumes primarily the disciplines of cognitive psychology and philosophy of mind. In practice, it is concerned with cognitive processes such as thinking, language, memory, problem solving, and creativity. One of two theoretical presuppositions is made, namely, that mind results from processes analogous to those used by computers or that it results from the parallel distributed processing of networks of connected units. Knowledge is derived from the observation of behaviour, including verbal behaviour, and from theoretical analyses.

A third domain of discourse is the experiential involving phenomenological, humanistic and transpersonal approaches to consciousness. This includes discussion of both subjective and private features of consciousness. Knowledge is derived from introspection as well as from the accounts of others concerning their experiences.

The first thing to note about these domains of discourse is that they are domains of discourse, not necessarily of phenomena. For example, ingestion of a psychedelic substance is a physiological event that has specific neural effects, such as the stimulation of S_2 receptor sites (Levinthal 1996), yet has perceptual and cognitive as well as experiential effects some of which can be profound and persist for years as in the case of seminary students given psilocybin prior to a Good Friday service (Doblin 1991). The second thing to note is the lack of any widely accepted links between these domains of discourse. In fact, there has been considerable attention drawn to the presence of the explanatory gaps between these domains with much debate concerning the inability to account for experience in physiological and computational terms. The third thing to note is that, while both bodies and experiences seem to be eminently accessible to an individual, the "middle" layer, the computational, may not exist except as a theoretical construct. "Mentalese", the purported formal language of the brain, along with the necessary axioms and rules of classical logic necessary for the processing of information in a manner analogous to that in a computer, has been called into question (Barwise 1986). And whereas parallel distributed processing models arose from neural networks through a process of abstraction, they have become so far removed from the actual biological processes they initially represented that what it is about the brain of which these are models remains to be seen (Hanson & Burr 1990, Smolensky 1988).

3 Definitions of Consciousness

What are we talking about when we talk about consciousness? There are four main categories of referents for the term consciousness as described in Barušs (Barušs 1987, 1990, 1992). At the most basic level, consciousness$_1$ is the characteristic of an organism in a running state that entails the registration, processing and acting on information as demonstrated by the organism's behaviour. Rather than getting into a debate about a minimum level of processing, consciousness$_1$ can be considered to be a variable. Similarly, rather than placing restrictions on the types of organisms that would qualify, such as larger mammals and humans, one can apply the term to any entity that meets these criteria keeping in mind that the manner in which they are instantiated may differ. That is to say, there is no reason to disqualify computers. It is important to note, with this and some of the other definitions of consciousness, that, in spite of some overlap, the concept of consciousness is not equivalent to that of awareness. In particular, whereas awareness is a passive property of an organism, consciousness has connotations of active agency as reflected in discussions concerning free will within consciousness studies.

Behavioural consciousness$_2$ designates the use of the term consciousness to refer to the explicit knowledge of one's situation, mental states or actions as opposed to lack of such awareness as demonstrated through one's behaviour. Subjective consciousness$_2$ refers to the stream of thoughts, feelings and sensations that occur for a person, some of which are more directly the focus of attention than others. It is subjective consciousness$_2$ that is most often identified as consciousness (Barušs 1990). The problem, of course, is that a person's experiences as such are private and inaccessible to the members of a scientific community. Behavioural consciousness$_2$ is the operationalization of subjective consciousness$_2$ so as to make it available for objective study. Conversely, having identified objective criteria for consciousness, one could infer subjective consciousness$_2$ from the presence of behavioural consciousness$_2$. In practice this applies to machines, whereby one would maintain that a machine that can pass the Turing test, that is to say, a machine behaviourally indistinguishable from a human with regard to its information-processing capabilities, is conscious. While this is clearly a logical error, some researchers have insisted that subjective consciousness$_2$ must be inferred from the presence of behavioural consciousness$_2$ (e.g., Lycan 1987).

There are those who have maintained that consciousness is more fundamental than indicated in the first three referents. That it is not that consciousness is the subjective stream of experience but that it is the sense of existence that allows for the possibility of there being a subjective stream at all. Usually this is accompanied by the contentions that there is a self for whom experience occurs and that states of pure consciousness without objects are possible. Thus, consciousness$_3$ is the sense of existence of the subject of mental acts. This is a definition given in subjective terms with no operational equivalent although it has sometimes been reified as an instance of subjective consciousness$_2$ (e.g., Natsoulas 1986).

4 Consciousness Surveys

It does not take long for someone interested in consciousness to notice the disparity of ideas about consciousness in the academic literature. These ideas appear to follow the beliefs of individual investigators, in particular, their fundamental beliefs along the material-transcendent dimension that underlies Western culture (Barušs 1990, 1992). In order to empirically test this contention, Robert Moore and I developed, through a number of stages, an instrument that could be used for measuring beliefs about consciousness and reality (Barušs 1990, Barušs & Moore 1989,1992).

In an initial study in 1986, we circulated a consciousness questionnaire to academics and professionals chosen on

the basis of the likelihood that someone from their discipline would write about consciousness in the academic literature (Barušs 1990). We received 334 completed copies of the questionnaire. The participants had a mean age of 44 years, 27% were women, 67% had obtained a doctorate, 42% were allied with psychology, 12% with physics, 6% with philosophy, while smaller numbers represented a variety of other disciplines (Barušs & Moore 1992). A material-transcendent dimension clearly emerged with items about reality and consciousness intertwined with one another.

The material pole of this dimension is characterized by agreement with statements that reality is physical in nature and that science is the proper way in which to know it. Consciousness is thought to emerge from neural activity or information processing, to always be about something, and is defined as consciousness$_1$ or behavioural consciousness$_2$. A conservatively transcendent position is defined by importance placed on meaning in life and adherence to traditional religious beliefs. Subjective aspects of consciousness are emphasized with subjective consciousness$_2$ and consciousness$_3$ as preferred definitions. Not surprisingly, consciousness is perceived to give meaning to life and to provide evidence of a spiritual reality. At the transcendent pole is an extraordinarily transcendent position whereby not only is the ontological hegemony of physical reality questioned but relegated to the status of a byproduct of consciousness. Respondents tending toward this position were more likely to claim to have had anomalous experiences such as out-of-body experiences, to believe in paranormal phenomena such as extrasensory perception and the continuation of life after death, and to find value in inner exploration. Universal consciousness is the goal of self-transformation as well as the key that makes the process of change possible. Rather than definitions that apply to the waking state of consciousness, altered states of consciousness are emphasized (Barušs 1990, 1992; Barušs & Moore 1989, 1992).

Using a more recent version of the initial instrument, called the Beliefs About Consciousness and Reality Questionnaire, a consciousness survey was conducted of participants at the scientific meeting Toward a Science of Consciousness 1996 'Tucson II'. Two hundred and twelve completed questionnaires were received of which 29% were from women and 56% from those who had earned a doctorate. The mean age of respondents was 50. A broad range of disciplinary categories from the natural sciences to arts and humanities was represented. The overall score of 18.3 on the global Transcendentalism scale was higher than that of 1.2 for the 1986 sample with a range of possible scores from S114 to 114. This may be due to a younger cohort for the 1996 study or to more transcendent beliefs of researchers with an actual rather than possible interest in consciousness. Those with an interest in neural correlates of consciousness tended to have low scores while those with an interest in phenomenology and culture had high scores. Thus the domains of discourse concerning consciousness are dominated by particular beliefs. Physiological aspects of consciousness are likely to be discussed by materialists while experiential aspects by transcendentalists (Barušs & Moore 1998).

While it is often thought that science should proceed without interference from the biases of the scientists carrying out the research, such is clearly not the case for consciousness studies. In particular, an investigator's contention that she has had anomalous experiences such as mystical or out-of-body experiences appears to incline her toward a transcendent position with its attendant emphasis on the primacy of consciousness. This dependence is not surprising given that the study of consciousness is concerned with subjective experiences that are accessible as such only to each investigator for herself (Barušs 1990, 1992, 1996).

5 Changes in Students' Beliefs

Given the importance of beliefs about consciousness and reality for the study of consciousness, a question arises concerning the conditions under which a person's beliefs about consciousness and reality would change. One situation in which they could change may be a university course in which transcendental issues are explicitly addressed as part of the course curriculum. Such a situation was presented in some of the classes taught by the author. Students' spontaneous comments concerning the courses had indicated that, in some cases, their beliefs had changed. Hence, in a continuation of our earlier research, Robert Moore and I decided to document those changes. It was hypothesized that students' beliefs would move in the direction of their instructors' beliefs.

For the 1995S96 and 1996-97 school years, students in my undergraduate Humanistic Psychology and Consciousness courses, taught at a small, Canadian, liberal arts, Catholic college, were given the Beliefs About Consciousness and Reality Questionnaire at the beginning of classes in September, around the time of the mid-year examination in December and again around the time of the final examination in April. Students in these classes were presented with data challenging materialist assumptions about the nature of reality and were required to understand the substance of transcendentalist arguments. Also during 1995-96 students in a Psychology of Creativity class taught by another instructor at the same institution and students in an Introductory Psychology course taught by yet another instructor at a separate but comparable small, Canadian, liberal arts, Catholic university were also administered the questionnaire using the same schedule. In addition, during the 1996-97 school year, students in my undergraduate Statistics for Psychology course were included in the study in the same manner as the others. In all cases, students were assured that their responses to the questionnaire would not be observed until all grades for

the course had been submitted.

Analysis of the reliability of the Transcendentalism scale of the Beliefs About Consciousness and Reality Questionnaire using all of the student data gives values for Cronbach's alpha of .88 (n = 220), .89 (n = 145) and .92 (n = 141) for each of the consecutive administrations. Thus, the instrument has good reliability when administered to students. Changes in students' beliefs between the initial and final administrations of the questionnaire for students who completed it on both of those occasions are given in Table 1.

Table 1

Within Subjects Analysis of Variance for Changes in Students' Scores on Transcendentalism Scale of Beliefs About Consciousness and Reality Questionnaire

Class	Years	*N*	Initial		Final		*F*
			M	*SD*	*M*	*SD*	
Humanistic & Consciousness	1995-96 & 1996-97	41	31.10	32.06	45.72	33.60	21.19*
Statistics	1996-97	24	24.25	21.66	33.96	24.31	24.01*
Instructor	1995-96	1	74.00		81.00		
Introductory	1995-96	57	13.51	18.65	14.16	19.15	0.13
Instructor	1995-96	1	45.00		43.50		
Creativity	1995-96	7	35.50	14.69	43.93	21.56	1.45
Instructor	1995-96	1	20.50		23.50		

*significance level $p < .0005$.

n refers to the number of participants
M refers to the mean score on the Transcendentalism Scale
SD refers to the standard deviation
F refers to the F test statistic for within subjects analysis of variance

While high to begin with, as expected, students' scores in my Humanistic Psychology and Consciousness classes moved in the transcendent direction during the school year. Students' scores in the Introductory Psychology course remained the same despite the fact that the instructor's scores were numerically higher than the class mean. Students' scores in the Psychology of Creativity class increased numerically although not to a statistically significant degree, despite class means numerically higher than the instructor's scores. Surprisingly, students' scores in my Statistics for Psychology course also moved in the transcendent direction even though there was no explicit discussion of transcendental issues in that class.

Clearly there are many influences on a students' beliefs of which an instructor's beliefs in a single course are only one possibility. The data indicate that there may be an institutional effect of moving students toward transcendent beliefs irrespective of the beliefs of single individual instructors at the particular college at which I and the Psychology of Creativity instructor teach. The effect was not observed at the university at which the Introductory Psychology course was taught. Alternatively, since the effect was only observed for courses that I teach it may be that I have an effect on students' beliefs without explicit discussion of transcendental issues that overrides other influences on students. In either case, what the data reveal is that beliefs about consciousness and reality can change during an undergraduate course, at least as measured by the Beliefs About Consciousness and Reality Questionnaire and that more research is needed to understand the reasons for the changes that occur.

This research has implications for the study of consciousness. What one talks about, what definitions one is willing to adopt and what attributes consciousness may have are tied to beliefs about consciousness and reality of the investigators doing the talking. In addition to anomalous experiences, education may influence the beliefs of investigators and hence appears to play a significant role in contemporary consciousness studies. Those entrained in materialist interpretations of reality

and those exposed to transcendentalist arguments may reflect the respective biases of their instructors in their research. This is an issue of fundamental importance for the study of consciousness.

6 References

[1] Baruss, I. (1987). Metanalysis of Definitions of Consciousness. *Imagination, Cognition and Personality,* 6, 4, p. 321-329.

[2] Baruss, I. (1990). *The Personal Nature of Notions of Consciousness*. Lanham, Maryland: University Press of America.

[3] Baruss, I. (1992). Contemporary Issues Concerning the Scientific Study of Consciousness. *Anthropology of Consciousness*, 3, 3 & 4, p. 28-35.

[4] Barušs, I. (1996). *Authentic Knowing: The Convergence of Science and Spiritual Aspiration.* Lafayette, Indiana: Purdue University Press.

[5] Baruss, I. & Moore, R. J. (1989). Notions of Consciousness and Reality. In J. E. Shorr, P. Robin, J. Connella and M. Wolpin (Eds.), *Imagery: Current Perspectives,* p. 87-92. New York: Plenum.

[6] Baruss, I. & Moore, R. J. (1992). Measurement of Beliefs About Consciousness and Reality. *Psychological Reports,* 71, p. 59-64.

[7] Barušs, I. & Moore, R. J. (1998). Beliefs About Consciousness and Reality of Participants at 'Tucson II'. *Journal of Consciousness Studies,* 5, 4, p. 483–496.

[8] Barwise, J. (1986). Information and Circumstance. *Notre Dame Journal of Formal Logic,* 27, 3, p. 324S338.

[9] Doblin, R. (1991). Pahnke's "Good Friday Experiment": A Long-term Follow-up and Methodological Critique. *The Journal of Transpersonal Psychology,* 23, 1, p. 1–28.

[10] Hanson, S. J. & Burr, D. J. (1990). What Connectionist Models Learn: Learning and Representation in Connectionist Networks. *Behavioral and Brain Sciences,* 13, 3, p. 471S489.

[11] Levinthal, C. F. (1996). *Drugs, Behavior, and Modern Society*. Boston, Massachusetts: Allyn and Bacon.

[12] Lycan, W. G. (1987). *Consciousness*. Cambridge, Massachusetts: MIT Press, Bradford.

[13] Natsoulas, T. (1986). Consciousness: Consideration of a Self-intimational Hypothesis. *Journal for the Theory of Social Behaviour*, 16, 2, p. 197S207.

[14] Smolensky, P. (1988). On the Proper Treatment of Connectionism. *Behavioral and Brain Sciences,* 11, 1, p. 1S23.

7 Acknowledgements

I thank Robert J. Moore for his collaboration in this research including the changes in students' beliefs described in the last section. I also thank Ted Wright and Diane Humphrey for administering the Beliefs About Consciousness and Reality Questionnaire to their classes during the 1995-96 academic year, Julienne Patterson for data entry and assistance with data analyses, King's College for research grants that were used to finance the changes in students' beliefs project and to all the students who chose to participate.

Recursion of Logical Operators and Regeneration of Discrete Binary Space

Jeremy Horne
15 Copper Hill Ct.
Durham, NC 27713
E-mail: jhorne1@cris.com

Keywords: logic, recursion, operators, order

Edited by: Anton P. Železnikar
Received: September 11, 1999 **Revised:** February 18, 2000 **Accepted:** March 15, 2000

Any discrete (closed) binary, or Boolean, space is recursive. That is, if the outputs of functions are repeatedly forward-fed into those functions, those outputs will present themselves again for processing. That is, the full functionality of an operator reproduces itself. Each of the 16 operators, or functions, in a two variable system is a self- maintaining (homeostatic) automaton in logical space. The homeostatic character of the function is displayed by that recursion. As larger binary spaces are comprised of the functions (partial or complete), functional recursion may open the way to analysing basins of attraction in spaces produced by the random concatenation of operators to reveal the character of pattern generation. Further, recursion of binary logical operators may have correlates in biological neural networks.

1 Synopsis

Any discrete (closed) binary, or Boolean, space is recursive. That is, if the outputs of functions are repeatedly forward-fed into those functions, those outputs will present themselves again for processing. That is, the full functionality of an operator reproduces itself. Each of the 16 operators, or functions, in a two variable system is a self- maintaining (homeostatic) automaton in logical space. The homeostatic character of the function is displayed by that recursion. As larger binary spaces are comprised of the functions (partial or complete), I suggest that functional recursion may open the way to analysing basins of attraction in spaces produced by the random concatenation of operators to reveal the character of pattern generation. Further, recursion of binary logical operators may have correlates in biological neural networks. Two issues emerge about how to approach pattern analysis in binary space.

First, the way each operator renders an environment displays its information processing character and efficiency. The character of a function's recursion is revealed by its outputs, and efficiency is measured by how rapidly an operator manages the complexity of the other functions it processes. (There is no issue of accuracy in such a measurement, since the outcome is deterministic.) Both character and efficiency may be components in discovering the sources of patterns in basins of attraction or deciphering what appears to be noise in digital space.

Second, prioritisation schemes in a parenthesis-free notation might be based on such an efficiency in order to measure overall computational efficiency in the system. Ranking of operators in a parenthesis-free (ungrouped) expression based on information processing efficiency also may be a means for exploring the basis of efficient logical thinking, or how fast a person can process logical alternatives to arrive at a correct conclusion.

2 The system's syntax – Structure of the function

There are 16 operators generated from a binary relationship schema, each operator being a function with a specific degree of complexity, as will be discussed informally below. We obtain descriptions of functions with the permutations of symbols, such as 0 and 1 (00, 01,10, and 11), resulting in a logical space (table of functional completeness) in the form of a 4 row by 16-column table, with each column headed by functions f_0 through f_{15}. I use the symbol set {0,1} to indicate columns that start with 0000 under the f_0 operator and end with 1111 under f_{15}. This would be 0 through F in hexadecimal.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Figure 1: Table of Functional Completeness.

All 16 relationships display how the first element is related to the second for each permutation of 0 and 1. For example, 0011 (the "0" often regarded as a "false," and 1 as "true") can be related to 0101 as 0111, and we call the relationship "or" (0 or 0 = 0, 0 or 1 = 1, etc.). Each function, then, is an ordered set of four elements. Note also that 0011 (or {0011}) is f_3 and 0101 is f_5. In standard truth table form using placeholders p and q for the functions f_3 and f_5, respectively and systematically displaying all four permutations of 0 and 1 in a binary relationship, f_7 is:

p	q	p or q
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2: Permutations of 0 and 1.

Each row of the function is a "deductive instance," where there is a description of a specific relationship between two points in space-time. The third row for example has the deductive instances of 1 and 0 to yield 1.

The functions, or operators, both process information in logical space, and are, themselves, units of information. Alpha characters are placeholders, or the variables, for functions, as in $p \vee (q \supseteq r)$, where the number of rows in the "truth table" rendition of the expression equals 2^n, n being the number of variables. With p * q, the number of rows required to describe a function completely equals four, where * is any function consisting of the four bits representing particular deductive instances. P * q * r requires eight rows; p * q * r * s needs 16 rows, and so forth. As soon as one determines the number of placeholders for values, the size of logical space is automatically determined, as well. However, the 4 rows by 16 columns logical space is the basic building block of these larger logical spaces, as may be seen by inspection. There are two types of logical space: pre-determined and sequentially ordered by the number of variables (as illustrated above), and space resulting from operations (such as a "truth table" computation). In a formal deductive proof, premises, including expressions of coupled functions, imply conclusions by the means of rules (axioms, inference rules, and equivalences). It must be borne in mind that every premise, intermediate expression, and conclusion in a proof is a function.

3 The system

The system in which the functions, or operators, perform is closed, and both functions and system seek to maintain integrity. Disorder, the breakdown in explicit placement of elements, results when the system accepts information not already found within its universe; certainty turns to probability. Deduction is a closed process, where any function or coupling of functions as a rule or rules, and entities upon which those rules act, yields a result predetermined by that coupling. (Recall that a function also is information.) In common parlance, "if the premises are true, the conclusion must be, as well." There are two aspects of deduction. First, there is system deduction, where an introduction of information outside the system, such as a function other than the ones within the system or an arbitrary extension of logical space with other functions disorders the system. A second aspect concerns maintaining the integrity of structures within the system. This could be called subsystem deduction.

4 Functional Recursion

If the operator is continuously "fed" two functions at a time without regard to previous output, there will result a logical space with a pattern (Wuensche and Kauffman, Chapter 5). Since each output is binary, it will be found as a column in logical space, hence contained within the system and deducible. It is not new information.

Emerging patterns from these ostensibly randomly coupled operations in a closed system suggest a way of discovering the nature of that reputed randomness. If we can place restrictions on how the information is given to the operator, we can see that the repetitive character of the operators within the space shows functional self-maintenance.

By feeding the outputs back to the four-digit operator as inputs, the outputs eventually will be repetitive, meaning that the function has stabilized. A systems analysis view provides a reason for this. Think of a binary relationship, such as p & q as being an atomic proof, that is, "{0011} and {0101}. Therefore {0001}." A deductive logic proof seeks stability. In the case of self-maintenance, the proof will accept information (other functions) as premises and process them, ultimately reaching a goal state of stabilization, where the output is a repetition of the input. Feeding this input back into the proof as premises simply repeats the proof. For both the function and proof, there is no longer produced any new information, there being be stabilization, the function(s) having reached the goal or deductive state. A full complement of information has been processed by the function. A system (be it a single operator interacting with other operators or a proof) that interacts with its environment and maintains itself in a state of equilibrium is called a homeostatic automaton. The next section describes its mechanics.

5 Structure and process of the binary homeostatic automaton

Keeping with our designation of p and q variables as placeholders for functions [$f_\bullet$(p, q)], the recursion, using f_2 as an example works in the following manner. (Note that each function is an ordered set.)

1. $f_2(f_3,f_5) = f_2$: {0010}({0011}, {0101}) = {0010}

2a. $f_2(f_2, f_5) = f_2$: {0010}({0010}, {0101}) = {0010} = {0010} The "p" half terminates, since the output of f_2 is a repetition of a previous output, f_2, and its reprocessing as an input obviously will result in another repetition of f_2.

2b. $f_2(f_3, f_2) = f_1$: {0010}({0011}, {0010}) = {0001} (Note that the order to be evaluated is f_3, f_2, and NOT f_2, f_3)

3a. $f_2(f_1,f_5) = f_0$: {0010}({0001}, {0101}) = {0010} = {0010} This half now is f_0, by virtue of 2b and continues on with $f2(f_0,f_5)$ and $f_2(f_5,f_0)$.

3b. $f_2(f_3,f_1) = f_2$: {0010}({0011}, {0001}) = {0010} This half terminates with f_1, by virtue of 2b.

The function is unstable ultimately for only four iterations.

A state diagram exhibiting its homeostatic behaviour may represent each operator. For example, these are the state diagrams and graphs for the f_7 and f_9 homeostatic functions.

		Iteration 1
+ ↑ Branchings ↓ -		$f_7(f_7,f_5) \rightarrow f_7$
	$f_7(f_3,f_5) \rightarrow f_7$	
		$f_7(f_3,f_7) \rightarrow f_7$

Figure 3: State Diagram for f_7.

Every time a set of p and q variables is used, there is created a branch, or divergence. In f_7 there are two branches, one for the f_7 function in the p placeholder, and the other for the f_7 in the q placeholder. The graph for f_7 is:

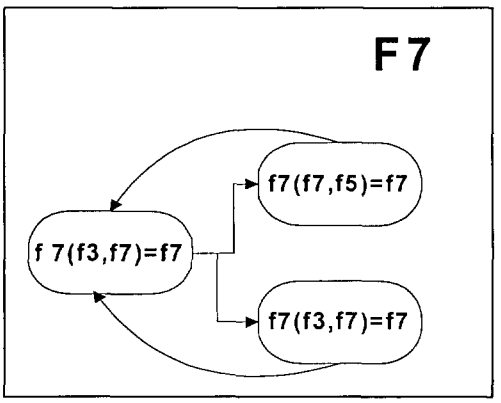


Figure 4: Graph for f_7.

The state diagram for f_9 is:

f_9

Branchings	Iteration 1	Iteration 2	Iteration 3
	$f_9(f_3,f_5) \rightarrow f_9$		$f_9(f_5,f_5) \rightarrow f_{15}$
$f_9(f_9,f_5) \rightarrow f_3$		$f_9(f_{15},f_5) \rightarrow f_5$	
			$f_9(f_3,f_5) \Rightarrow f_9$
	$f_9(f_3,f_3) \rightarrow f_{15}$		
			$f_9(f_3,f_5) \rightarrow f_9$
		$f_9(f_3,f_{15}) \rightarrow f_3$	
			$f_9(f_3,f_3) \rightarrow f_{15}$
	$f_9(f_5,f_5) \rightarrow f_{15}$		
$f_9(f_3,f_9) \rightarrow f_5$			
	$f_9(f_3,f_5) \rightarrow f_9$		

Figure 5: State Diagram for f_9.

In f_9, there is one branch for the p side of the diagram, but there are ultimately three branches, or divergences on the q side. The graph for f_9 is:

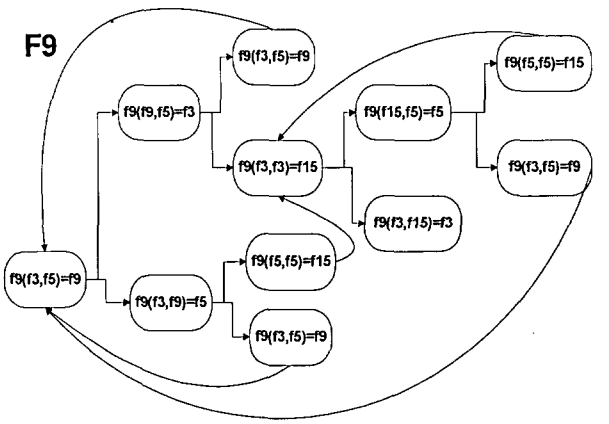


Figure 6: Graph for f_9.

Each of the 16 functions is recursive, i.e., $f_*(f_3, f_5)$. Furthermore any function over any other two functions, $f_*(f_n,f_p)$ is recursive, as well, because there can be only a maximum of 16 iterations (only 16 unique functions) on any branch before there is a repetition of one of the previous. Perforce, even if there are 16 unique outputs on a branch, there is repetition on the 17th iteration. Because of this, no function can have more than sixteen iterations.

Diagrammatically,

For the p side:	For the q side:
F*(fp, fq) = fa	F*(fp, fq) = fa
F*(fa, fq) = fb	F*(fp, fa) = fb
F*(fb, fq) = fc	F*(fp, fb) = fc
....	....
Max F*(fp≤16, fq) before there is repetition	Max F*(fp, fq≤16) before there is repetition

Figure 7: Maximum Iterations is 16.

The number of times a function receives input before it repeating the output (number of iterations) and how much information or logical space the operator consumes (number of "sub-functions" generated by each iteration) will vary with the function. So, f_{13} may process f_3 and f_5 recursively for three iterations and terminate, but f_8 might take seven iterations. The same initial information is processed, but the state diagram shows the other areas of logical space involved, thus giving the function an "anatomical description" of its attempt to gain homeostasis. The two vector components are the number of branches, or divergences, at each iteration (node creations) and number of iterations required to reach equilibrium, the point where the function is stabilized and starts repeating its outputs. Each operator has a different complexity and can be ordered according to its vector descriptions. In terms of computational efficiency, a function has to process a quantity of information (or it only has to process that much less information) to maintain itself as a homeostatic automaton. (To symbolically describe a function, one might take the Cartesian product of the differentials of both the X and Y components of the function.) A question is whether the same efficiency exists for the function acting recursively over any two initial functions other than f_3 and f_5.

6 Recursion of Discrete Binary Spaces

A discrete binary space is an n-dimensional bounded area in which each component assumes exclusively either of two values, conditions, or states. Thus, each of the 16 functions discussed above occupy a discrete binary space. In larger discrete spaces, randomly generated bit streams display "basins of attraction," or repeating functions, according to Wuensche and Kauffman. Inasmuch as such basins are repetitive and are comprised of binary functions, it would reasonable to ask if the recursive characteristics of those functions discussed above would also apply to these spaces and be useful in understanding the repetitions within those spaces.

Essentially, the methodology is the same for examining any n-dimensional space recursively as is with the individual functions. Following is an informal presentation of this approach to spaces of two or more dimensions. It is not meant not to be a complete discussion but illustrative. There are three ways in which a discrete binary space can be shown to be recursive.

- First, since the space is composed of one or more of the 16 functions and because functions are recursive, we see that the space, then, is recursive.
- Second, because $f_*(f_p, f_q)$ is recursive, any column of the space can act over the next two columns recursively.
- Third, the environment for an n-dimensional space consists of the permutations of bits of that space. For example each function as a one dimension has as its environment, 2^4, or 16 permutations. An i by j matrix has 2^{ij} permutations as its environment. To examine the recursive character of a space:

1. f*(AS, PS) = NS, where f* is one of sixteen functions, AS is the initial space being examined, PS is one of the possible spaces, and NS is the resulting space. In matrix notation, where AS, PS, and NS are matrices: |AS| * |PS| = |NS|.

Recursively:

2A. f*(NS, PS) = NS, or |NS| * |PS| = |NS| first branch

2B f*(PS, NS) = NS, or |PS| * |NS| = |NS| second branch for each of the 16 functions, resulting in $16(2^{ij})$ generations.

7 Character of Attractors – Starting point for analysis

Whenever a function produces an output previously produced, there is begun a cycle that has been previously iterated. This repetition point forms the basis of an attractor. Every function operating over two other functions results in one of more emerging attractors. While it seems that logical operators are randomly coupled to produce patterns, it would be interesting to see what patterns emerge if the operators were coupled according to an empirically determined scheme based upon a specific ordering of operators. (Horne, 1997)

A concept of information processing efficiency of each operator may be used to disassemble the recursive process within the larger spaces and the boundaries between patterned and chaotic space. To do this requires affixing an order of operators based upon a standard, a subject for further research. A starting point would be to examine how many iterations it would take for each function to process inputs f_3 and f_5. From the above, the functions could be ranked according to the number of iterations required to process f_3 and f_5. Thus, f_0, f_1, f_3, f_5, f_7, , and f_{15} might be ranked co-equally and most efficient in maintaining themselves,, since each has only one iteration. Those functions having two iterations, f_{10} and f_{12}, would be the next most efficient, and so forth. Taking into account the number and type of branches could be a factor in a function's complexity and bear on how it processes its environment.

8 Ramifications and applications of theory – avenues for exploration

A discrete binary space is comprised of elements having one of two possible conditions, a factor of great import. Any phenomenon that can be digitised is subject to

recursive analysis. Both ordered and unordered phenomena can be digitised and set in a discrete binary space. Sunspot activity, waves, and background "noise" are candidates. Every piece of computer code, whatever the language, can be reduced to machine language, 0s and 1s, meaning that every computer program is a discrete binary space. To say that a discrete binary space is recursive is to say, also, that any computer program is recursive and can reproduce itself.

How would one determine whether there is order, or pattern, within any discrete binary space? There are attraction points found within autonomous random Boolean network (BN) state-space (Kauffman 1993, Chapter 5). With respect to ordering in Boolean complexity, it can be demonstrated that numerous random couplings of operations result in patterns resembling those of cellular automatons (Wuensche 1993, passim). By "dissecting" these spaces by functional recursion, it may be possible to discover the nature of any attractors and dispel any notion that the patterns were generated "randomly." Too, spaces without any discernible pattern might also be analysed with recursion.

9 Summary

Any discrete binary space can repeat itself because it is composed of one or more of the 16 binary logical functions that are recursive. The reason fed-back four bit functions repeat themselves is the same as for the more than four bit functions, as the latter are composed of the former. (Partial functions – three or fewer digits - are repeatable, albeit the graphs are more complicated. Each possible outcome has to be graphed. {001}, for example would require graphs for {0010} and {0011}.) A discrete binary space as a homeostatic automaton processes its initial environment, thereby producing outputs that, *ipso facto*, change the character of the environment. The discrete space processes the changed environment, and, in the course of doing so, tries to maintain itself. After a number of iterations, if the initial function (or space) is homeostatic, it will stabilize, evidenced by repeating outputs characteristic of how it processes the environment. Discrete binary spaces may be analysable with the recursion method; a larger space reputedly randomly generated and containing a pattern may be analysable with functional recursion and may not be random at all

A number of actual situations arise where binary grouping is at issue in analysing spaces. Some operators may be more efficient than others in the way they process information and maintain themselves in the environment. Logic and automatons seem to describe how neural pulses behave (Hameroff and Penrose. passim) and a question is whether operational ordering would make a difference in how binary patterns emerge from the surface of neuronal microtubules. Basins of attraction may bear a relation to cellular automatons, or electroencephalograms (EEG), as suggested by Wuensche. (Wuensche 1993, p.11) EEGs may be correlated with temporal coding in neural populations, since neural nets operate in groups to process information, and a temporality enters in information processing. (Fetz p. 1901) If temporality is important in neuronal information and logical operations can be correlated to EEGs, then, logical operations are temporally bound. Further, neural networks "...exhibit emergent properties such as ... generation of distinct outputs depending on input strength and duration, and self-sustaining feedback loops." (Bhalla and Iyengar , p. 381) If mapped to biological neural structures, binary recursion as discussed above, becomes immediately relevant. To add to the space-time theme, I have argued elsewhere that the values of 0 and 1 are representations of wave function collapse, similar to "truth tables." (Horne.1997)

As can be seen above, there is a philosophical and theoretical side to the structure of binary logic. It is a bit (not only a pun) more than a crude device used for ordinary language translation.

10 References

[1] Bhalla, Upinder S. and Iyengar, Ravi (1999) "Emergent Properties of Networks of Biological Signaling Pathways" Science, 283, p. 381-387.

[2] Fetz, Ebherd E. (1997) "Temporal Coding in Neural Populations?" *Science*, 278, p. 1901-1902.

[3] Hameroff S. and Roger Penrose (1996) "Orchestrated Reduction of Quantum Coherence in Brain Microtubules: A Model for Consciousness." *Toward a Science of Consciousness: The First Tucson Discussions and Debates.* Eds. S.R. Hameroff et al. MIT Press. Cambridge, MA, p. 507-540.

[4] Kauffman S. (1993) *The Origins of Order* Oxford University Press. New York.

[5] Horne J. (1997) "Logic as the Language of Innate Order in Consciousness" *Informatica,* Vol. 22, No.4, pp. 675-682.

[6] Hayes P. J., Novak G. S. \& Lehnert W. G. (1992) ACM Forum - In Defence of Artificial Intelligence. *Communications of the ACM*, 35, 12, p. 13-14.

[7] Wuensche A. (1993) "The Ghost in the Machine. Basins of Attraction of Random Boolean Networks" in *Cognitive Science Research Papers, CSRP 281* University of Sussex at Brighton.

ERK'2000
Electrotechnical and Computer Science Conference
Elektrotehniška in računalniška konferenca
September 21–23, 2000

Conference Chairman
Baldomir Zajc
University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 349, Fax: 386 1 4264 630
E-mail: Baldomir.Zajc@fe.uni-lj.si

Conference Vice-chairman
Saša Divjak
University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 260, Fax: 386 1 4264 647
E-mail: Sasa.Divjak@fri.uni-lj.si

Program Committee Chairman
Jurij Tasič
University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 260, Fax: 386 1 4264 630
E-mail: Jure.Tasic@fe.uni-lj.si

Programe Committee
Tadej Bajd
Genevieve Baudoin
Gerry Cain
Saša Divjak
Janko Drnovšek
David J. Evans
Matjaž Gams
Ferdinand Gubina
Marko Jagodič
Karel Jezernik
Jadran Lenarčič
Drago Matko
Miro Milanovič
Andrej Novak
Nikola Pavešić
Franjo Pernuš
Kurt Richter
Borut Zupančič

Publications Chairman
Franc Solina
University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: 386 1 4768 389, Fax: 386 1 4264 647
E-mail: franc@fri.uni-lj.si

Advisory Board
Rudi Bric
Damjan Dittrich
Miloš Urbanija

Call for Papers

for the nineth **Electrotechnical and Computer Science Conference ERK'2000**, which will be held on 21–23 September 2000 in Portorož, Slovenia.

The following areas will be represented at the conference:

- *electronics,*
- *telecommunications,*
- *automatic control,*
- *simulation and modeling,*
- *robotics,*
- *computer and information science,*
- *artificial intelligence,*
- *pattern recognition,*
- *biomedical engineering,*
- *power engineering,*
- *measurements,*
- *didactics.*

The conference is organized by the **IEEE Slovenia Section** together with the Slovenian Electrotechnical Society and other Slovenian professional societies:

- Slovenian Society for Automatic Control,
- Slovenian Measurement Society,
- SLOKO–CIGRE,
- Slovenian Society for Medical and Biological Engineering,
- Slovenian Society for Robotics,
- Slovenian Artificial Intelligence Society,
- Slovenian Pattern Recognition Society,
- Slovenian Society for Simulation and Modeling,
- Slovenian Language Technologies Society.

Authors who wish to present a paper at the conference should send two copies of their final camera-ready paper to Baldomir Zajc, Faculty of Electrical Engineering, Tržaška 25, 1001 Ljubljana. The paper should be max. four pages long. More information on `http://www.ieee.si/erk00/`

Time schedule: Camera-ready paper due: *July 24, 2000*
Notification of acceptance: *End of August, 2000*

Call for Paper
International Multi-Conference
Information Society - IS'2000

17 – 19 October, 2000
Slovenian Science Festival
Cankarjev dom, Ljubljana, Slovenia

Invitation

You are kindly invited to participate in the "New Information Society - (IS'2000)" multi-conference to be held in Ljubljana, Slovenia, Europe, from 17–19 October, 2000. The multi-conference will consist of eight carefully selected conferences.

Basic information

The concepts of information society, information era, infosphere and infostress have by now been widely accepted. But, what does it really mean for societies, sciences, technology, education, governments, our lives? What are current and future trends? How should we adopt and change to succeed in the new world?

IS'2000 will serve as a forum for the world-wide and national community to explore further directions, business opportunities, governmental European and American policies. The main objective is the exchange of ideas and developing visions for the future of information society. IS'2000 is a standard scientific conference covering major recent achievements. Besides, it will provide maximum exchange of ideas in discussions, and concrete proposals in final reports of each conference.

The multi-conference will he held in Slovenia, a small European country bordering Italy and Austria. It is a land of thousand natural beauties from the Adriatic sea to high mountains. In addition, its Central European position enables visits to most European countries in a radius of just a few hours drive by car. The social programme will include trips by desire and organised trips to Skocjan or Postojna caves. Coffee breaks, the conference cocktail and dinner will contribute to a nice working atmosphere.

Call for Papers

Deadline for paper submission: 15 July, 2000

Registration fee is 100 US $ for regular participants (10.000 SIT for participants from Slovenia) and 50 US $ for students (3.500 SIT for Slovenian students). The fee covers conference materials and refreshments during coffee-breaks.

More information

For more information visit
`http://ai.ijs.si/is/indexa00.html` or contact `milica.remetic@ijs.si`.

The multi-conference consists of the following conferences:

- Information society and governmental services
- Media in information society
- Education in information society
- Warehouses and data mining
- Development and reingeeniring of information systems
- Production systems and technologies
- Cognitive science
- Language technologies.

International Programme Committee:

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, Scotland
Vladimir Fomichov, Russia
Alfred Inselberg, Izrael, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, Japan

Call for Papers

International multidisciplinary conference on Emergence, Complexity, Hierarchy, Organisation, to be held from the 31st of July to the 4th of August 2000 at the University of South Denmark, in Odense. All details are to be found on the conference website: http://www.hum.edu.dk/center/filosofi/emergence

THE MINISTRY OF SCIENCE AND TECHNOLOGY OF THE REPUBLIC OF SLOVENIA

Address: Trg OF 13, 1000 Ljubljana,
Tel.: +386 61 178 46 00, Fax: +386 61 178 47 19.
http://www.mzt.si, e-mail: info@mzt.si
Minister: Lojze Marinček, Ph.D.

Slovenia realises that that its intellectual potential and all activities connected with its beautiful country are the basis for its future development. Therefore, the country has to give priority to the development of knowledge in all fields. The Slovenian government uses a variety of instruments to encourage scientific research and technological development and to transfer the results of research and development to the economy and other parts of society.

The Ministry of Science and Technology is responsible, in co-operation with other ministries, for most public programmes in the fields of science and technology. Within the Ministry of Science and Technology the following offices also operate:

Slovenian Intellectual Property Office (SIPO) is in charge of industrial property, including the protection of patents, industrial designs, trademarks, copyright and related rights, and the collective administration of authorship. The Office began operating in 1992 - after the Slovenian Law on Industrial Property was passed.

The Standards and Metrology Institute of the Republic of Slovenia (SMIS) By establishing and managing the systems of metrology, standardisation, conformity assessment, and the Slovenian Award for Business Excellence, SMIS ensures the basic quality elements enabling the Slovenian economy to become competitive on the global market, and Slovenian society to achieve international recognition, along with the protection of life, health and the environment.

Office of the Slovenian National Commission for UNESCO is responsible for affairs involving Slovenia's co-operation with UNESCO, the United Nations Educational, Scientific and Cultural Organisation, the implementation of UNESCO's goals in Slovenia, and co-operation with National commissions and bodies in other countries and with non- governmental organisations.

General Approaches – Science Policy

Educating top-quality researchers/experts and increasing their number, increasing the extent of research activity and achieving a balanced coverage of all the basic scientific disciplines necessary for:
- quality undergraduate and postgraduate education,
- the effective transfer and dissemination of knowledge from abroad,
- cultural, social and material development,
- promoting the application of science for national needs,
- promoting the transfer of R&D results into production and to the market,
- achieving stronger integration of research into the networks of international co-operation (resulting in the complete internationalisation of science and partly of higher education),
- broadening and deepening public understanding of science (long-term popularisation of science, particularly among the young).

General Approaches – Technology Policy

- promotion of R&D co-operation among enterprises, as well as between enterprises and the public sector,
- strengthening of the investment capacities of enterprises,
- strengthening of the innovation potential of enterprises,
- creation of an innovation-oriented legal and general societal framework,
- supporting the banking sector in financing innovation-orientated and export-orientated business
- development of bilateral and multilateral strategic alliances,
- establishment of ties between the Slovenian R&D sector and foreign industry,
- accelerated development of professional education and the education of adults,
- protection of industrial and intellectual property.

An increase of total invested assets in R&D to about 2.5% of GDP by the year 2000 is planned (of this, half is to be obtained from public sources, with the remainder to come from the private sector). Regarding the development of technology, Slovenia is one of the most technologically advanced in Central Europe and has a well-developed research infrastructure. This has led to a significant growth in the export of high-tech goods. There is also a continued emphasis on the development of R&D across a wide field which is leading to the foundation and construction of technology parks (high -tech business incubators), technology centres (technology-transfer units within public R&D institutions) and small private enterprise centres for research.

R&D Human Potential

There are about 750 R&D groups in the public and private sector, of which 102 research groups are at 17 government (national) research institutes, 340 research groups are at universities and 58 research groups are at medical institutions. The remaining R&D groups are located in business enterprises (175 R&D groups) or are run by about 55 public and private non-profit research organizatios.

According to the data of the Ministry of Science and Technology there are about 7000 researchers in Slovenia. The majority (43%) are lecturers working at the two universities, 15% of researchers are employed at government (national) research institutes, 22% at other institutions and 20% in research and development departments of business enterprises.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of **Slove**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 61000 Ljubljana, Slovenia
Tel.:+386 61 1773 900, Fax.:+386 61 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in .eps format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Rudi Murn, Informatica, Institut Jožef Stefan, Jamova 39, 61111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than five years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Informatica WWW:

http://ai.ijs.si/informatica/
http://orca.st.usm.edu/informatica/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Vladimir Bajič, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Jerzy Brzezinski, Marian Bubak, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Netiva Caftori, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, David Cliff, Maria Cobb, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Deborah Dent, Andrej Dobnikar, Sait Dogru, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Jozo Dujmović, Pavol Ďuriš, Johann Eder, Hesham El-Rewini, Warren Fergusson, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Hugo de Garis, Eugeniusz Gatnar, James Geller, Michael Georgiopolus, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Inman Harvey, Elke Hochmueller, Jack Hodges, Rod Howell, Tomáš Hruška, Don Huch, Alexey Ippa, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Sabhash Kak, Li-Shan Kang, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Kevin Korb, Gilad Koren, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Phil Laplante, Bud Lawson, Ulrike Leopold-Wildburger, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Matija Lokar, Jason Lowder, Kim Teng Lua, Andrzej Małachowski, Bernardo Magnini, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Daniel Memmi, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Hari Narayanan, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, William C. Perkins, Warren Persons, Mitja Peruš, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Gustav Pomberger, James Pomykalski, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Wolf Rauch, Peter Rechenberg, Felix Redmill, David Robertson, Marko Robnik, Ingrid Russel, A.S.M. Sajeev, Bo Sanden, Vivek Sarin, Iztok Savnik, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, William Spears, Hartmut Stadtler, Olivero Stock, Janusz Stokłosa, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiři Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Wieslaw Traczyk, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Zygmunt Vetulani, Olivier de Vel, John Weckert, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Zonling Zhou, Robert Zorc, Anton P. Železnikar

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Volume 24 Number 2 June 2000 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics