Volume 10, Number 2 **ISSN 1524-7252**

ACADEMY OF INFORMATION AND MANAGEMENT SCIENCES JOURNAL

An official Journal of the

Academy of Information and Management Sciences

Sharad K. Maheshwari
Editor
Hampton University

Editorial and Academy Information
may be found on the Allied Academies web page
www.alliedacademies.org

The Academy of Information and Management Sciences is a subsidiary of the Allied Academies, Inc., a non-profit association of scholars, whose purpose is to support and encourage research and the sharing and exchange of ideas and insights throughout the world.

Whitney Press, Inc.

Printed by Whitney Press, Inc.
PO Box 1064, Cullowhee, NC 28723
www.whitneypress.com

The *Academy of Information and Management Sciences Journal* is owned and published by the Entrepreneurship Group, LLC, PO Box 2689, Cullowhee, NC 28723, USA, (828) 293-9151, FAX (828) 293-9407. Those interested in subscribing to the *Journal*, advertising in the *Journal*, submitting manuscripts to the *Journal,* or otherwise communicating with the *Journal*, should contact the Allied Academies' Executive Director at info@.alliedacademies.org.

ACADEMY OF INFORMATION AND MANAGEMENT SCIENCES JOURNAL EDITORIAL BOARD MEMBERS

ACADEMY OF INFORMATION AND MANAGEMENT SCIENCES JOURNAL

CONTENTS

LETTER FROM THE EDITOR

Welcome to the *Academy of Information and Management Sciences Journal,* the official journal of the Academy of Information and Management Sciences. The Academy is one of several academies which collectively comprise the Allied Academies. Allied Academies, Incorporated is a non-profit association of scholars whose purpose is to encourage and support the advancement and exchange of knowledge.

The editorial mission of the *AIMSJ* is to publish empirical and theoretical manuscripts which advance the disciplines of Information Systems and Management Science. All manuscripts are double blind refereed with an acceptance rate of approximately 25%. Manuscripts which address the academic, the practitioner, or the educator within our disciplines are welcome. And, diversity of thought will always be welcome.

Please visit the Allied Academies website at www.alliedacademies.org for continuing information about the Academy and the *Journal*, and about all the Allied Academies and our collective conferences. Submission instructions and guidelines for publication are also provided on the website.

Sharad K. Maheshwari
Hampton University

www.alliedacademies.org

Editor's Note: In a previous issue of *AIMSJ*, Volume 9, Number 2, 2006, we published a paper by Dr. Chang Won Lee, Jinju National University, entitled "Development of Web-Based Decision-Support System for Business Process Re-Engineering in a Health Care System." We published that manuscript by mistake. The author had submitted it for publication in the *Journal of Strategic E-Commerce*, and we did publish it there in Volume 4, Number 2, 2006. We apologize for the error and we would like to emphasize that the double publication was in no way the fault or responsibility of the author.

STUDENTS' PERSONALITY TYPE AND CHOICE OF MAJOR

Bill McPherson, Indiana University of Pennsylvania
Scott Mensch, Indiana University of Pennsylvania

ABSTRACT

This study sought to determine if there was a relationship between personality type, using the Myers-Briggs Type Indicator (MBTI), and information technology students' choice of major (Business Information Systems, Computer Information Systems, and Management Information Systems). Results indicated there existed a significant relationship between major and personality type. The recommendations provide direction and strategies for information technology educators to increase enrollments in the respective majors using personality type.

"Like most people, I remember very well the year I chose my major. In fact, I made the decision three times." Dr. David Brownlee, Professor of Art History, University of Pennsylvania (2006).

RATIONALE FOR THE STUDY

In reading any information about employment trends one is inundated with the theme of a need for technology workers. In his February 2004 State of the Union speech, President Bush stressed that "as technology transforms the way almost every job is done, America becomes more productive, and workers need new skills…We must respond by helping more Americans gain the skills to find good jobs in our new economy" (Spalding, 2005). Spalding (2005) stated

The technology field continues to have some of the country's fastest-growing occupations. The Bureau of Labor Statistics projects increases of 82 percent in the number of positions for network and computer systems administrators; and a 97 percent rise for computer support specialists, between 2000 and 2010. It also predicts job growth for software engineers of up to 100 percent, depending on area of expertise, for the same time period.

Major companies and government officials have agreed there is a continuing need for technology workers and recruiting from universities and colleges is still one of the best ways to find competent technology workers as well as workers in many other fields (A. Schieli, personal interview, July 6, 2006). Many colleges and universities have technology majors, whether it is computer science (CIS), management information systems (MIS), or business information systems (BIS). Universities and colleges must continue to recruit for these fields to support the future

demand in the workforce. Students are attracted to various majors for a number of reasons: faculty, reputation, future careers, coursework, etc., but personality type is one variable that has been overlooked by students when choosing a major.

The use of personality testing is certainly not new to the field of information technology. Morrill (2005) stated one of the most important aspects of being a manager is finding and hiring people who will work best within the organization. This is an often overlooked process when hiring information technology (IT) personnel and needs to be observed during the hiring process.

Given this information and review of literature, pursuing a study that examines a student's choice of major in relation to personality type is appropriate.

STATEMENT OF PROBLEM

An appreciation and understanding of students' personality types and preferences for majors is needed. Research has revealed that there needs to be a growing pool of candidates who are technologically savvy, educated, and skilled. Employment data illustrates the vast job opportunities in fields related to technology. Students will embrace occupational choices where employment is in demand. In order to meet future employment demands in the workplace, there needs to be an increasing number of students choosing technology as their majors. Once specific personalities are typed, the results could be used by IT faculty to successfully match and attract students by creating a profile plan to link various personality types with specific majors. According to the University of Minnesota website in an article entitled: A Major Decision…A Workbook for Choosing and Changing your Major (Pascarella and Terezini, 1999) offers,

Nationally, three out of four entering students express some uncertainty about their choice of major, while between one and two-thirds change majors. Recent studies suggest that it is typical for undergraduates to test out four or five majors before arriving at a decision

Given these facts, IT faculty can pursue these students for recruitment into their respective IT fields. The results of the study could provide information concerning the various facets of specific personality types and provide educators with a schema to build strategies, methods, and systems to magnetize students to the IT fields. For this study, technology majors will be defined as students who are pursuing a degree in BIS (called various names within university and college settings) CIS and MIS.

OBJECTIVES

This research study will focus on the personality types of technology majors. Certain other related factors will also be considered. The following research questions will be addressed:

- What are the personality types of technology majors?
- Are technology majors' personality types different or similar?

- What MBTI classifications are most/least popular among technology majors?

The following primary null hypothesis guided the study:

H0. There will be no tested differences among technology students personality type and choice of major.

H1. There will be tested differences among technology students' personality types and choice of major.

LITERATURE REVIEW

Technology, Careers, and Demand

In today's fast paced world, the demand for professionals in technology is greater than ever. Salaries in the IT field are increasing in hopes of attracting employees. Scott Melland (2006), CEO of Dice, a technology job site, states

> *The tech job market has rarely been hotter. According to the Bureau of Labor Statistics, unemployment in the IT sector ranges form 2-3 percent, depending on the job title—less than half the national average. The number of openings posted on technology job site Dice.com is up 18 percent compared with last year and 63 percent since 2004.*

To further support this demand Microsoft is working with the U.S. Department of Labor to strengthen the U.S. work force and boost the ability of the nation's companies to compete in a global economy by donating cash, software, and a curriculum to help provide training in technology skills to adults (Passman, 2006).

Technology majors have many options after graduation; there are plenty of colleges and universities that can meet their needs. Tynan (2006) states in order to attract and retain top IT talent, IT hiring managers are in overdrive, devising new methods to lure candidates in a technology workers' market. Faculty in IT majors need to adhere to this formation to lure candidates into their majors.

College Major

Various academic institutions define Management Information Systems (MIS), Business Information Systems (BIS), and Computer Information Systems (CIS) in unique ways. The U.S. Bureau of Labor Statistics (2005) projects tremendous growth and high salaries for those seeking

careers in computer-related industries, and academic institutions such as James Madison University, Indiana University of Pennsylvania, Bloomsburg University, Moorehead State University, and the University of Whitewater, offer programs geared towards those who have an interest in the computer industry.

Management Information Systems (MIS)

In reviewing universities offering undergraduate degrees in MIS it was found that students should expect to receive specialized training in both technical skills and business knowledge. Major curriculum components gathered from course descriptions, curriculum guides, and syllabi include database management, systems analysis, and the design and management of front-end / back end business applications.

Computer Information Systems (CIS)

In an effort to entice students and to broaden their program the Universities studied defined Computer Science programs as a CIS degree. In reviewing universities offering undergraduate degrees in Computer Science it was found that students should expect a fundamental education with sufficient understanding of basic principles and concepts in computer science to solve computational problems. Major curriculum components gathered from course descriptions, curriculum guides, and syllabi include programming, hardware support, and computer graphics.

Business Information Systems (BIS)

In describing this major, universities vary in the names used to describe their program. Business Technology Support, Organizational/Office Systems, and End User Technology are just a few titles for this major. However, the course descriptions associated with this major are very similar. For the purpose of this paper this major will be titled Business Information Systems (BIS). In reviewing Universities offering undergraduate degrees in BIS, it was found that students should expect to fill roles as computing specialists, technology coordinators and trainers, and network administrators. Major curriculum components gathered from course descriptions, curriculum guides, and syllabi include telecommunications, Web design, and computer based applications.

In researching the degrees offered in IT it is found that there are sufficient students to support these programs. However astute educators must be able to promote each program to better identify potential students and ensure that the correct technology major is chosen. Each technology degree encompasses specific components of IT. Identifying these differences and matching them to the student's personality type will help ensure the students' growth and success in their chosen field of study. This will subsequently lead to a greater pool of employees for the various IT fields.

Personality Type

Measuring an individual's personality dispositions and preferences, the Myers-Briggs Type Indicator (MBTI) is one of the most popular and widely used personality measurement instruments (Davis, 2006; Handler, 2006; Irani & Scherler & Harrington, 2006; and Johnson, 2006). It is designed to identify the personality preferences of an individual along four dichotomous indices. The MBTI classifies people, based on their self-reporting behavior, preferences, and value judgments, into dichotomous categories along each of the four interlocking dimensions: extroversion-introversion, sensation-intuition, thinking-feeling, and judgment-perception. Myers and McCaulley (1989) explain each index as:

E-I Index: The E-I index reflects whether an individual is extroverted or introverted. Extroverts (E) are oriented primarily to the outer world of people and things. Introverts (I) are oriented more toward the inner world of concepts and ideas.

S-N Index: The S-N index reflects how a person chooses to gather information or perceive the world. Sensing (S) people prefer to rely most heavily on the five senses to observe facts or happenings. Intuitives (N) prefer to perceive the world as possibilities, meanings and relationships, relying more on hunches or insights rather than on the five senses.

T-F Index: The T-F index reflects how a person prefers to make judgments or decisions. A person who prefers using the thinking (T) function makes decisions based more upon logic or objective fact. A person who prefers to use the feeling (F) function makes decisions based more on personal or social values and subjective beliefs.

J-P Index: The J-P index indicates how an individual prefers to deal with the world. A person who prefers judgment (J) uses one of the judging functions (either thinking or feeling), usually seeks closure, and is more orderly and planned in behavior. A person preferring perception (P) uses one of the perceptive functions (either sensing or feeling) and is usually more spontaneous, open, and adaptable in lifestyle.

This classification is an adaptation of Carl Jung's theory of conscious psychological type. In addition, many researchers (Allen, 1988; Kiersey and Bates, 1984; and Myers and McCaulley 1989) have supported the application of the MBTI to a variety of educational settings. Jung's theory and the utilization of the MBTI are appropriate means of studying the relationship between subjects' personality types and their preferred majors.

INSTRUMENTATION

The population sample for the study was seven universities in the U.S. which offer a degree in either BIS, CIS, or MIS. Students enrolled in the aforementioned majors from those universities/colleges were the population for the study. Participation was voluntary. One research instrument was utilized for the collection of data in this study.

Form G of the Myers-Briggs Type Indicator (MBTI) was administered to determine the personality type of the technology majors in the study. The MBTI is a self-reported, forced choice indicator designed to assess personality type and is based on Carl Jung's personality type theory.

This study followed a descriptive research design using survey methods with statistical treatments. The design was a cross-sectional survey. Babbie (1990, p. 65) stated that the cross-sectional design is the most frequent used study design. Babbie (1990) supports the use of this type of survey when

> *...data are collected at one point in time from a sample selected to describe some larger population at that time. Such a survey can be used not only for purposes of description but also for the determination of relationships between variables at the time of study* (p. 62).

When employing survey research, one must be aware of the advantages and disadvantages of this type of research. Frankfort-Nachmias and Nachmias (1996), when discussing the survey method offered, "…its major advantages are lower costs, relatively small biasing error, greater anonymity, and accessibility. Its disadvantages are a low response rate, opportunity for probing, and the lack of control over who fills out the questionnaire" (p. 248).

In order to minimize the disadvantages of using the survey methods, "the design method" (TDM) suggested by Dillman (1978) was used as a guide. Dillman (1978) defined the TDM as

> *...consisting of two parts. The first [part] identifies each aspect of the survey process that may affect either the quality or quantity of response and to shape each of them in such a way that the best possible responses are obtained. The second [part] organizes the survey efforts so that the design intentions are carried out in complete detail* (p. 12).

Using Dillman's TDM will help to minimize the problems of response quality and quantity. The data in this descriptive study was collected using survey procedures as described by Dillman (1978).

DATA ANALYSIS

Through the use of the Statistical Package for the Social Sciences for Windows (SPSS+ for Microcomputers, release 4.0), statistical tests were performed on the data from the scale. Descriptive and comparative analyses were made. Throughout this study a decision rule (i.e., level of significance) will be used at the accepted convention of $p \leq .05$ (Borg and Gall, 1989; Kerlinger, 1986).

The chi-square analysis is the most popular method of showing the statistical significance, or independence, based on the cross-tabulation table that has been created. If the chi-square analysis shows that the variables are not significant, the null hypothesis can not be rejected because there is no relationship between the variables. If the chi-square analysis shows that the variables are significant, then the null hypothesis can be rejected because a relationship between the variables has been established (Connor, 2006).

Chi-square statistics were used in the present study to measure the significance of Students' personality type and choice of major, specifically in relation to technology majors. The threshold in reporting a relationship as statistically significant was set at $p < .05$. All cross-tabulation charts measuring the ethics and compliance plan awareness by management level and by gender in the present study also used the chi-square statistics at $p < .05$.

Cross-tabulation analysis is an effective means to identify a relationship between the variables being studied and is a common tool used by many researchers. Cross-tabulation charts record the frequency of the survey responses in a two-dimensional chart. Statistical significance is established or rejected using the chi-square analysis (Connor, 2006).

FINDINGS

Demographics

Surveys were sent to 448 participants which resulted in a response rate of 248 (55%). Due to missing cases related to various variables, the participant number in each variable studied will vary and not add up to 248 or 100%. Table 1 indicates the frequencies of personality types in the study. Table 2 indicated that there was an even distribution between the three information technology majors (BIS, CIS, and MIS). Table 3 indicated various grade levels. Table 4 provides information concerning gender.

Table 1: Frequencies -- Actual MBTI			
		Frequency	Percent
Valid	ISTJ	34	13.7
	ESTJ	25	10.1
	ISTP	20	8.1
	ESFJ	20	8.1
	ENTP	18	7.3
	ENTJ	18	7.3
	ESTP	17	6.9
	ISFJ	16	6.5
	ENFP	15	6.0
	INTP	12	4.8
	INTJ	11	4.4
	INFP	9	3.6
	ISFJ	7	2.8
	INFG	6	2.4
	ENFJ	5	2.0
	ESFP	3	1.2
	Total	236	95.2
Missing	999	12	4.8
Total		248	100.0

Table 2: Frequencies – Majors			
		Frequency	Percent
Valid	BIS	68	27.4
	CIS	68	27.4
	MIS	74	29.8
	Total	210	84.7
Missing	999	38	15.3
Total		248	100.0

Table 3: Highest Grade Completed			
		Frequency	Percent
Valid	Junior	120	48.4
	Senior	63	25.4
	Sophomore	26	10.5
	Freshman	5	2.0
	Total	214	86.3
Missing	999	34	13.7
Total		248	100.0

Table 4: What is your major?				
Gender	BIS	CIS	MIS	Valid Percent
Female	38	10	29	36.8
Male	30	57	45	63.2
Total	68	67	74	100.0

Table 5 provides information concerning race.

Table 5: Frequencies - Race				
		Frequency	Percent	Valid Percent
Valid	Caucasian/White	156	62.9	82.5
	Asian/Pacific Islander	20	8.1	10.6
	Other	7	2.8	3.7
	African American	6	2.4	3.2
	Total	189	76.2	100.0
Missing	999	59	23.8	
Total		248	100.0	

PRIMARY HYPOTHESIS

The study sought to determine if a relationship existed between personality type and choice of major. It was discovered that a relationship did exist with a significance level of .001. The null hypothesis is rejected and the alternate is accepted. To explain the found significance in this study,

a description of the dominant personality type is provided, literature on each personality type is discussed, and associated careers are listed. After exploring each major with the information above, the information clearly illustrates traits which match various careers based on the curricula of the major.

Table 6 provides the three most frequent personality types related to BIS. This table indicates that the dominant preferences in these categories are E, S, T and J. The three dominant personality types are described below:

ESTJ's are briefly described as group oriented, leaders, outgoing, social, do not like being alone, and formal. Literature suggests their favored careers are business consultants, managers, businesspersons, office managers, public relations manager, and business analysts.

ESTP's are briefly described as social, group oriented, do not like to be alone, and are good at getting people to have fun. Literature suggests their favored careers are marketing specialists, business managers, consultants, technicians, and business people.

ESFJ's are briefly described as values organized religion, values relationships and families over intellectual pursuits, group oriented, content, and positive. Literature suggests their favored careers are business consultants, human resources managers, human resources directors, and social services directors.

Table 6: Frequencies -- Actual MBTI and BIS

		Frequency	Percent	Valid Percent
Valid	ESTJ	8	11.8	11.8
	ESTP	7	10.3	10.3
	ESFJ	7	10.3	10.3

Table 7 provides the three most frequent personality types related to CIS. This table indicates that the dominant preferences in these categories are I, S, T and J. The three dominant personality types are described below:

ISTJ's are briefly described as private, loner tendencies, organized, detail oriented, would rather be friendless than jobless, observer, focused, finisher, punctual, private, follows the rules, logical, analytical, and hard working. Literature suggests their favored careers are data analysts, scientists, researchers, office workers, office managers, account managers, analysts, systems analysts, technicians, and network administrators.

INTJ's are briefly described as loners, more interested in intellectual pursuits than relationships or family, would rather be friendless than jobless, does not tend to like most people, socially uncomfortable, can be lonely, and prepared. Literature suggests their favored careers are systems analysts, researchers, computer scientists, software designers, computer programmers, and biotechnologists.

ISTP's are briefly described as hidden, private, loner tendencies, and dislikes leadership. Literature suggests their favored careers are technicians, computer scientists, software engineers, software developers, systems analysts, computer animators, and data analysts.

Table 7: Frequencies -- Actual MBTI and CIS

		Frequency	Percent	Valid Percent
Valid	ISTJ	14	20.6	20.6
	INTJ	9	13.2	13.2
	ISTP	8	11.8	11.8

Table 8 provides the three most frequent personality types related to MIS. This table indicates that the dominant preferences in these categories are E, S, T and J. The three dominant personality types are described below:

ISTJ's are briefly described as private, loner tendencies, organized, detail oriented, would rather be friendless than jobless, observer, focused, finisher, punctual, private, follows the rules, logical, analytical, and hard working. Literature suggests their favored careers are data analysts, scientists, researchers, office workers, office managers, account managers, analysts, systems analysts, technicians, and network administrators.

ESTJ's are briefly described as group oriented, leaders outgoing, social, do not like being alone, and formal. Literature suggests their favored careers are business consultants, managers, businesspersons, office managers, public relations manager, and business analysts.

ESFJ's are briefly described as values organized religion, values relationships and families over intellectual pursuits, group oriented, content, and positive. Literature suggests their favored careers are business consultants, human resources managers, human resources directors, and social services directors.

To support the study findings, Myers & McCaulley (1989) offered this about ISTJs. "They often choose jobs where their talents for organization and accuracy are rewarded" (p. 27). The material reviewed shows that all CIS majors displayed introverted (I) attitudes which enjoy "solitude and privacy" (p. 27).

To support the study findings, academic websites, such as Bloomsburg, University of Central Florida, and Indiana University of Pennsylvania, provided this about MIS:

> …that newer areas such as networking, cybersecurity, and user/manager involvement in the global business environment are integrated with the traditional skills of programming, analysis and design, database development, various architectures, and application development. Microprocessor technology, the mainframe environment, and client server applications are included, as well as an emphasis on business

computing issues such as profitability, budgeting, collaboration, and project management.

The MIS curriculum includes coverage of computer programming, database design and implementation, networks and data communications, systems analysis, systems implementation, managerial decision making, and managerial aspects of organization information systems (University of Central Florida).

Many of these functions described above are very in tune with the descriptors in the ISTJ personality types.

In researching academic web sites, peer reviewed articles, and textbooks it was found that CIS majors, which are primarily ISTJ personality types, will be drawn to programs that offer opportunities in the fields of computer science.

Table 8: Frequencies -- Actual MBTI and MIS

		Frequency	Percent	Valid Percent
Valid	ISTJ	9	12.2	12.2
	ESTJ	8	10.8	10.8
	ESFJ	8	10.8	10.8

In studying the top three personality types in each major, several interesting trends were discovered. All BIS students were found to display extrovert personality types while CIS majors were found to display introvert personality types. (See Tables 6 and 7). MIS students were found to have both introvert and extrovert personality types. (See Table 8). These findings show that extrovert personality characteristics associated with BIS students match tasks such as working with people, training, and end-user support that are defined by their major. CIS introvert personality types however are geared towards technology and not the businesses and people who run them. MIS is a mixture of both technology and business which explains both introvert and extrovert personality types being present. A review of any job descriptions and titles for BIS and CIS would concurred with their favored careers.

All BIS and MIS majors displayed sensing personality types as compared to CIS majors which showed students as having both sensing and intuition characteristics. (See Tables 6, 7, and 8). This may be explained as both BIS and MIS majors are involved in teaching technology from a global business perspective which involves the people behind the technology that drives business practices. CIS is more involved with the software and hardware behind the business rather than the actual people in the business operations. The descriptors about their personality types maybe stereotypical, but are consistent with career choice.

Both BIS and MIS were found to exhibit thinking and feeling personality types while CIS students were only found to display thinking personality types. (See Tables 6, 7, and 8). This can be explained by institutions defining CIS with a technical and scientific orientation and not encompassing the tasks of associating people in the processes. BIS and MIS programs prepare their students to work with both people and technology. This involves being flexible, open, and organized while having the ability to directly support the end users' technical issues.

CIS and BIS students were found to show both judging and perceiving personality types while MIS students only exhibited judging personality types. (See Tables 6, 7, and 8). MIS majors are expected to work in the fields of project planning, analysis, and design which all involve having a structured and organized approach and involve being task oriented. BIS majors must be flexible and open to new experiences as working directly with employees and groups invites new and unexpected experiences. CIS majors work in a more casual and open environment not associated directly with the business and people who operate the technology.

Given the vast amount of detail, information, and relationships, it comes as no surprise to see that a relationship would occur between technology majors' personality type and major. The analysis of the chi-square probability of $p < .001$ resulted in accepting the alternate hypothesis that there exists tested differences among technology majors personality types and their choice of major.

ANCILLARY HYPOTHESES

The study further sought to look at the two primary variables of major and personality type to see if relationships existed between these two variables and the ancillary variables of race, gender, and grade level.

In looking at major and the ancillary variables, a relationship existed between major and grade level (.000), major and gender (.000) and major and race (.051).

Major and Grade Level

In analyzing the highest grade completed among the students surveyed several interesting trends were discovered. Of the students 2% were freshmen, 48.4% were juniors, 10.5% were sophomores, and 25.4% were seniors. No students who declared their major as MIS were freshman, 13% were sophomores, 78.3% were juniors, and 8.7% were seniors. CIS had no freshman, 8.8% sophomores, 55.9%, juniors, and 35.3% seniors. BIS had 7.4% freshman, 14.7% were sophomores, 38.2% juniors, and 39.7% seniors. Each technology major experienced a dramatic increase between the sophomore and junior years. This is most likely the result of undecided students declaring their major after their sophomore year. While both MIS and CIS had over 35% of the students being seniors, less than 9% of MIS students were in their final year. This decrease could be the result of several factors such as the prerequisites, class sequence, and transfers. Further research is needed to address this issue.

Major and Gender

Males accounted for the majority of the study with 63.2% as compared to 36.8% of females. CIS and MIS both accounted for the majority of male students. CIS had the most male students, 57, with only 10 female students. MIS contained 45 male students as compared to only 29 female students. BIS accounted for the least amount of male students with 30 and had more female students with 38 in the major. This gender reversal within the BIS major may be the result of the academic institutions that were surveyed historically defining their major with other terms such as office systems and training.

Major and Race

Caucasians accounted for the majority of information technology students as 83.2% were white, compared against 16.8% of non-Caucasian technology students. BIS accounted for the most Caucasians within the technology degrees studied with 93.6% Caucasians and no Asian students. (See Table 4). CIS majors were compiled of 81% Caucasian and 11.9% Asians. (See Table 5). MIS accounted for the most Asian students with 16.7% and 77.8% Caucasian students. (See Table 6). An argument can be made that Asian students are drawn to majors that involve technology and are less interested in the training and direct end-user support associated with the BIS major. In addition, fearful of communication skills some international students who major in technology lean towards computer science because of the stereotype or misperception that this degree is about working with things and not people.

In looking at personality type and the ancillary variables, a relationship existed between personality type and gender (.000); and personality type and race (.001).

Personality Type and Gender

The relationship between personality type and gender are statically significant at the $p < .05$ level or better. When studying personality type and gender the three most common personality types for women were ESFJ (18), ESTJ (9), and ENTJ (6). The three most common personality types for males were ISTJ (22), ISTP (16), and ENTP (15).

Personality Type and Race

The differences between personality type and each major are statically significant at the $p < .05$ level or better. When looking at personality type and race it was found that Caucasians accounted for a majority of those studied. The three most common personally types for Caucasians were found to be ISTJ (23), ESTJ (15), and ESFJ (15). These findings mirrored the top three personality types of MIS majors. African Americans, who only had 5 respondents, were most

commonly found to be INTJ (2). Several personality types (ISTJ, ISFJ, ESTJ, and ENFJ) were all found to have 1 respondent. Asian/Pacific Islander were found most likely to exhibit ENTJ (3) personally traits. Several personality types (INTJ, INFJ, ESTJ, ENFJ, ISTP, and ENFJ) were all found to have 1 respondent. Only 6 of those polled were defined as other. The top two respondents who could not be defined as African Americans, Asian/Pacific Islander, or Caucasians were found to be INFJ (2) and ENTP (2). ESFP and ENFJ were both found to have 1 respondent. As stated earlier when looking at major and race, Caucasians were the dominant race who participated in the study. The results of the relationship between personality type and race are not a surprise given the pool of participants. In addition, the MBTI has typically been normed and studied with traditional white populations. This indicates that the MBTI may not be the most suitable way of determining the personality type of participants with diverse racial and cultural backgrounds.

SUMMARY

The current and future marketplaces indicate that technology workers will be needed to fulfill large vacancies. The U.S. Department of Commerce has forecasted that by 2006 technology positions will account for half of those employed in the workforce. Currently almost half of those employed in technology positions are employed by organizations that are not traditionally thought of as technology based companies such as supermarkets, financial institutions, and police forces. The U.S. Bureau of Labor Statistics has also shown that eighty percent of fastest growing careers our based in technology fields.

Given the results of the study, it is imperative that educators attract more students into technology majors and ensure that are successful in completing their degree. To meet this critical shortage in the workplace, college educators must make use of every available resource to attract students to IT majors. This study provides some valuable information and resources to use "personality type" as a method for recruitment and retention of students in IT majors.

Academic instructions struggle with retention and proper advising dilemmas. Identifying relationships between variables such as race and gender, which have never been studied before, may help aid in the improvement of retention and placement issues. Additional research between variables such as personality type, major, and GPA may also assist when recruiting technology students to ensure that they are in the correct major. Replication of this study in urban and suburban centers, as well as historically dominated African American or Asian American universities, will better help understand the role that a students personality type plays in their ability to choose the correct IT major.

REFERENCES

Allen, R. F. (1988). *The relationship between learning style and teaching style of secondary teachers in South Central Kansas*. Unpublished doctoral dissertation, Kansas State University, Manhattan, Kansas, 2.

Babbie, E. (1990), *Survey Research Methods*. Belmont, CA: Wadsworth Publishing.

Bloomsburg University of Pennsylvania. (2006). Business Information Systems. Retrieved September 20, 2006, from http://www.bloomu.edu/prospective/programs/ois.php

Bloomsburg University of Pennsylvania. (2006). Computer and Information Science. Retrieved September 20, 2006, from http://www.bloomu.edu/prospective/programs/cis.php

Bloomsburg University of Pennsylvania. (2006). *Management Information Systems*. Retrieved September 20, 2006, from http://www.bloomu.edu/prospective/programs/mis.php

Borg, W.R. & Gall, M.D. (1989). *Educational research* (5th ed.). New York, NY: Longman, Inc.

Brownlee, D. (2006). Choosing a College Major. Retrieved October 18, 2006 from http://www.college.upenn.edu/curriculum/major_choosing.html

Chambers, S.M. (2003). Personality indicators and emergency permit teachers' willingness to embrace technology. Retrieved December 2003 from http://findarticles.com/p/articles/mi_m0FCG/is_3_30/ai_108836886

Clark, C.B. & Wheeler. (1994). *Journal of Research on Computing in Education.* Vol. 26 Issue 3, p358, 13p.

Connor-Linton, F. (2006). Chi Square Tutorial Retrieved May. 1, 2007 from http://www.georgetown.edu/faculty/ballc/webtools/web_chi_tut.html

Grindler, M.C., & Straton, B.D. (1990). Type indicator and its relationship to teaching and learning styles. *Action in Teacher Education*, 7, 31-34.

Davis, K. (2006). United States: Personality Tests: Proceed with Caution. Retrieved Sept. 25, 2006 from http://www.mondaq.com/article.asp?articleid=42544&searchresults=1&login=true

Dillman, D.A. (1978). *Mail and telephone surveys: The total design method*. New York: John Wiley and Sons.

Frankfort-Nachmias, C. & Nachmias, D. (1996). *Research Methods in the Social Sciences*, 5th ed. New York: St. Martin's Press.

Hadfield, W. (2005). Psychometric tests help to identify IT bosses' skills. *Computer Weekly;* 8/2/2005, p31-31, 1/5p. Retrieved 11 December 2005 from http://web33.epnet.com/

Handler, C. (2005). Alphabet soup: A cautionary note about using Myers Briggs. Retrieved Sept. 25, 2006 from http://www.personalitytesting.com/Articles/Note_Using_Myers_Briggs.htm

Indiana University of Pennsylvania. (2006). *Business Technology Support*. Retrieved September 20, 2006, from http://www.eberly.iup.edu/ts/DegreesBachelorTST.shtm

Indiana University of Pennsylvania. (2006). Computer Science. Retrieved September 20, 2006, from http://www.iup.edu/registrar/catalog/Natscimath/compsci.shtm

Indiana University of Pennsylvania. (2006). *Management Information Systems*. Retrieved September 20, 2006, from http://www.iup.edu/registrar/catalog/ecob/MISandDecSci.shtm

Irani, T., Telg, R., Scherler, C., & Harrington, M. (2003). Personality type and its relationship to distance education students' course perceptions and performance. The Quarterly Review of Distance Education, 4(4), 445-453.

James Madison University. (2006). Computer Information Systems. Retrieved September 20, 2006, from http://www.jmu.edu/cis/wm_library/CIS_Undergraduate_Info_06_07.pdf

James Madison University. (2006). Computer Science. Retrieved September 20, 2006, from http://www.cs.jmu.edu/abouttheprogram.html

Johnson, T. (2006). Can your personality get you hired or fired? Retrieved Sept. 21, 2006 from http://abcnews.go.com/GMA

Jung, C. G. (1989). *Psychological types*, (5th ed.). London, UK: Routedge and Raud.

Kerlinger, F. N. (1986). *Foundations of behavioral research*, (3rd ed.). New York, NY: Holt, Rinehart, and Winston.

Kiersey, D. & Bates, M. (1984). *Please understand me: Character and temperament types* (5th ed.). Del Mar, CA: Prometheus Nemesis Books.

Know Your Type. (2005). Myers-Briggs. Retrieved October 6, 2006, from http://www.knowyourtype.com/

Maher, K. (2006). Latest Hiring Increase Is Good News for Grads. *The Wall Street Journal Online.* Retrieved October 19, 2006 from http://www.collegejournal.com/columnists/thejungle/20040607-jungle.html

Melland, S. (2006). Executive order: Attract and retain top IT talent. Retrieved October 19, 2006 from http://www.infoworld.com/article/06/09/18/38FEjobstalentwar_1.html?s=feature

Moorehead University. (2006). Business Information Systems. Retrieved September 20, 2006, from http://www.moreheadstate.edu/is/index.aspx?id=3397

Moorehead University. (2006). Computer Information Systems. Retrieved September 20, 2006, from http://www.moreheadstate.edu/is/index.aspx?id=3398

Morrill, D. (2005). Hiring for Culture as well as technical skill in Information Technology. Retrieved October 19, 2006 from http://www.infosecwriters.com/text_resources/pdf/Hiring_DMorrill.pdf

Myers, I. B. (1962). *Myers-Briggs Type Indicator manual*. Palo Alto, CA: Consulting Psychologists Press.

Myers, I. B. (1975). *Manual: Myers-Briggs Type Indicator*. Palo Alto, CA: Consulting Psychologists Press.

Myers, I. B. & McCaulley, M. H. (1989). *Manual: A guide to the development and use of the Myers-Briggs Type Indicator*, 15th ed. Palo Alto: CA: Consulting Psychologists Press.

Nickels, W.G., McHugh, J.M., & McHugh S.M. (2005) *Understanding Business* (Customized 7th ed.). New York: McGraw-Hill.

Pascarella, E.T. & Terezini, P.T. (1999). A Major Decision…A Workbook for Choosing and Changing You Major. Retrieved September 18, 2006 from http://ucs1.ucs.umn.edu/www/majorworkbook.html

Passman, P. (2006). Microsoft, Department of Labor Team Up to Train Technology Workers. Retrieved May 1, 2006 from http://pnnonline.org/article.php?sid=6565

Spalding, E. J. (2004). New opportunities for America's technology workers. *National IT Apprenticeship System from CompTIA answers President's call to create new opportunities for America's technology workers.* Retrieved May 1, 2006 from http://www.comptia.org

Team Technology. (2005). Myers Briggs. Retrieved October 6, 2006, from http://www.teamtechnology.co.uk/tt/t-articl/mb-simpl.htm

Tynan, D. (2006). Executive order: Attract and retain top IT talent. Retrieved October 19, 2006 from http://www.infoworld.com/article/06/09/18/38FEjobstalentwar_1.html?s=feature

U. S. Bureau of Labor Statistics. (2005). Career Guide to Industries: Computer Systems Design and Related Services. Retrieved October 19, 2006 from http://www.bls.gov/oco/cg/cgs033.htm

University of Central Florida. (2006). What is MIS? Retrieved December 2, 2006 from http://www.bus.ucf.edu/mis/public/downloads/pdf/what_is_mis.pdf

Whitewater University. (2006). Information Technology Infrastructure**.** Retrieved September 20, 2006, from http://www.uww.edu/cobe/departments/itbe/plan.html

Whitewater University. (2006). Management Computer Systems. Retrieved September 20, 2006, from http://www.mcs.uww.edu/viewUndergraduateProgram.do

AN EXPERIMENTAL DESIGN COMPARISON OF FOUR HEURISTIC APPROACHES FOR BATCHING JOBS IN PRINTED CIRCUIT BOARD ASSEMBLY

Susan K. Williams, Northern Arizona University
Michael J. Magazine, University of Cincinnati

ABSTRACT

The goal of the printed circuit board (PCB) job-batching problem is to minimize the total manufacturing time required to process a set of printed circuit board jobs on a pick-and-place machine. Specifically, to determine which jobs should be processed with the same setup so that the total number of setups is reduced without increasing the required processing time such that the reduction in setup time is offset. Since PCB manufacturers assemble thousands to millions of boards each year, even modest time savings per job are useful.

In this paper, we reports the results of an experiment designed to compare four heuristic approaches to solve the PCB job-batching problem: cluster analysis, a bin-packing approach, a sequencing genetic algorithm, and a grouping genetic algorithm. We developed these heuristic approaches in previous work. However, based on that work, there was not overall best heuristic. These results show that the cluster analysis and bin-packing approaches have fast execution time but do not find optimal solutions while the two genetic algorithms are slower but often find optimal solutions. Results describe the problem characteristics for which each heuristic performs best. Based on the results described here, a user can decide which heuristic is most appropriate based on PCB job characteristics and execution time requirements.

(Heuristic algorithms for combinatorial optimization; PCB assembly)

INTRODUCTION

We performed an experiment to compare the performance of four heuristic approaches to solve the printed circuit board (PCB) job-batching problem. This problem attempts to minimize the manufacturing time (both setup and processing time) for a set of PCB jobs on a pick-and-place machine. Magazine et al. (2002) have shown that this problem is NP-hard so heuristics must be used for difficult problems. In previous work (Williams and Magazine, forthcoming.), we

developed heuristic families to solve the PCB-JB problem. None of the heuristics performed best on all problems. In this paper, we describe an experiment and results to determine the problem characteristics for which a heuristic performs better than the other heuristics. Results were analyzed to determine the significant factors affecting solution quality and execution time.

The four heuristics that were compared are: 1) the *clustering heuristic* is based on a set of techniques adapted to our problem from the cluster analysis literature 2) the *bin-packing heuristic* is analogous to the best-fit-decreasing bin-packing algorithm 3) the *GASPP (Genetic Algorithm Shortest Path Problem) heuristic* is a sequencing genetic algorithm that uses a shortest-path algorithm for fitness evaluation. This heuristic searches the space of all job sequences 4) the *GGA (Grouping Genetic Algorithm) heuristic* searches the space of all partitions of a set of jobs.

The printed circuit board is a subassembly in most electronic products. In 1998, 72.5% of all PCBs produced were utilized by 3 major industries: computers, telecommunications, and consumer electronics. PCB production increased by 5% in 1998 to become a $35 billion industry (Nakahara 1999) and by the year 2000 was a $42.1 billion industry (Nakahara 2001).

In the printed circuit board manufacturing process, the assembly stage can be a system bottleneck (Ahmadi and Kouvelis 1999). During assembly, components such as resistors, capacitors, and integrated chips are placed onto the PCB. PCB assembly has evolved from a labor-intensive process to a highly automated process in which a pick-and-place machine inserts most of these components. According to Mody et al. (1992) a typical shop has 25-30 machines exceeding $1.5 million dollars in value. At Hewlett-Packard (HP), the capital investment of a single assembly line can exceed $2 million dollars (Jain et al. 1996).

The high capital investment combined with the competition among PCB manufacturers creates an environment requiring production efficiency (Mody et al. 1992). In addition, planning for assembly is a highly complex problem with many interrelated decisions. Due to the complexity of planning and the increased pressure for efficiency, companies are working to improve the planning process (Depuy et al. 2001).

One difficulty in achieving an efficient process is that a PCB shop may produce a high variety of boards in relatively low volume thus requiring frequent machine setups. The setup time for the pick-and-place machines can be lengthy. In the high mix environment at HP “it was not unusual to find that over 50% of the production time was spent in setup” (Jain et al. 1996 p. 843)

Magazine and Polak (2003) introduced and formulated this problem as an integer programming problem. For the special case of a pre-determined job sequence, they show that the problem can be solved efficiently as a shortest path problem. Cohn et al. (2001) have developed a branch and price algorithm to optimally solve the PCB-JB problem. Problems with up to 32 jobs were successfully solved to optimality. Some instances with 60 jobs were also successfully solved. However, not all problems of size 32 could be solved and the execution time varied from seconds to days.

In Williams and Magazine (forthcoming), we developed four heuristic families to solve the PCB-JB problem. For each family, several variations were created and tested on a variety of PCB-

JB problems. The variation with the best average performance was selected from each family. Both the clustering heuristic and the bin-packing heuristic construct a solution to the PCB-JB problem. The remaining two heuristics are genetic algorithms (GAs) that stochastically search a solution space for the best solution. None of the four heuristics outperformed the other for all test problems. Therefore, to determine which heuristics is likely to be most successful for a given problem, we have designed an experiment and present the results here.

The results from the experiment show that all four heuristics find high-quality solutions to the PCB-JB problem. If a single heuristic must be selected to solve all types of PCB-JB problems, we recommend the clustering heuristic as it finds the average best solution and is fastest. However, if we can select from the set of four heuristics knowing problem characteristics, our recommendation is:

1) If execution time is not crucial:
   - ♦ For problems with a small number of jobs, use the GGA heuristic.
   - ♦ For problems with a large number of jobs and little variation in the number of boards in a job, use the clustering heuristic.
   - ♦ For problems with a large number of jobs and high variation in the number of boards in a job, use the bin-packing heuristic.

2) If fast execution time is required (such as an on-line algorithm):
   - ♦ For problems with low variation in the number of boards per job, use the clustering heuristic.
   - ♦ For problems with high variation in the number of boards per job, use the bin-packing heuristic.

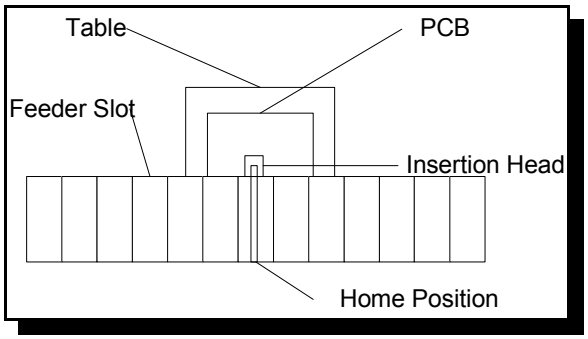
These four heuristics then are useful for two purposes:
- ♦ To determine good-quality solutions for PCB-JB problems that are too large or difficult for optimal algorithms.
- ♦ To provide good starting solutions to improve the performance of optimal algorithms on the PCB-JB problem.

In Section 2, we describe the PCB-JB problem in more detail and discuss related work. We introduce the best variation from each of the heuristic families in Section 3. Section 4 describes the comparison study, including the experimental design, the analysis, and a summary of the results. Finally, in section 5 we provide the recommendations for using the heuristics to solve the PCB-JB problem.

PROBLEM DESCRIPTION

During assembly, components are placed on the PCB using a pick-and place machine (Figure 1). The various components to be inserted onto a PCB are installed in the feeder slots during setup. A board is placed on the X-Y table which moves to align the insertion head with the next location for component insertion. To process a board, the head moves from its home position to the feeder slot that contains the next component type to be inserted, retrieves the component, and travels back to its home position for component insertion. During component retrieval, the table is concurrently positioning the board at the next insertion site. The table movement requires less time than component retrieval and is ignored. The process continues until all component types allocated to this machine and required by this PCB are inserted.

Figure 1: Pick-and-place machine



A *job* is an order for a specific type of PCB. At the beginning of a planning period, a set of jobs, $\boldsymbol{K}$, is waiting to be processed on a pick-and-place machine. The number of jobs in the planning set is K. The pick-and-place machine inserts components from the set of component types, $\boldsymbol{N}$, with cardinality N.

A job, $k\hat{I}\boldsymbol{K}$, is described by a board profile, R^k, and a job size, b^k. R^k is an N-vector, (r_1^k, r_2^k, r_3^k, ..., r_N^k), containing the number of each component type required for job k. (The board profile also typically contains the placement locations of the component. However, this is not relevant to our work.) The *job size* is the number of identical boards in the order for job k. The *job profile* is the total number of each component type required to process the job, b^kR^k.

A *batch,* $\boldsymbol{X}$, is a subset of $\boldsymbol{K}$ containing one or more jobs that will be processed using the same machine setup. There are $C=2^K-1$ batches that can be created from planning set, $\boldsymbol{K}$. The *batch*

profile, R^X, contains the number of components required by all jobs in batch $\boldsymbol{X}$, $R^X = (r_1^X, r_2^X, r_3^X, \ldots, r_N^X) = \left(\sum_{k \in X} b_1^k r_1^k, \sum_{k \in X} b_2^k r_2^k, \sum_{k \in X} b_3^k r_3^k, \ldots, \sum_{k \in X} b_N^k r_N^k \right)$.

The batch profile can vary significantly from the individual job profiles.

The operating characteristics of pick-and-place machines vary considerably and significantly affect problem formulation. The second author observed the operation of pin-thru-hole (PTH) pick-and-place machines. Though these machines are an older technology, they are still relevant. Nakahara (2003) estimated that China's PTH capacity was 31,600 km^2 panel/year in 2002 and would increase to 40,200 km^2 panel/year in 2003-04. Our assumptions for analysis with regard to the operating and technology characteristics of the pick-and-place machine were observed by the second author while working with a computer manufacturer.

1) After each component retrieval, the insertion head returns to its home position in order to insert the component onto the PCB.
2) Each component type is assigned to one feeder slot and all feed slots are the same width. Due to (1), there is no advantage to assigning a component type to more than one feeder slot.
3) No partial setups are performed. Each setup is a "full tear-down". Many PCB manufacturers prefer to perform a full tear down after each board is processed in order to minimize operator error. (Ahmadi et al. 1988)
4) The time required to setup the machine is constant and therefore independent of the set of jobs that will be processed. This is a result of (3). To perform a complete setup, the operator removes all feeder slots, loads them with the appropriate components, and re-installs the feeder slots.
5) The number of feeder slots on the pick-and-place machine is the same as the cardinality of the set of component types. This is not a limiting assumption because if the number of component types is larger than the number of feeder slots then the additional component types are installed by another pick-and-place machine, by a second pass through the same machine, or manually. If there are fewer component types than feeder slots, then the job profile will indicate the number of components of this type as 0 and will assign the component type to the most distant feeder slot. Therefore, there is no loss of generality by assuming that there are *N* component types for each PCB and *N* feeder slots on the pick-and-place machine.
6) Retrieval/insertion time increases as the distance from the feeder slot to the home position increases.
7) The component tapes installed in the feeder slots have unlimited capacity. For some pick-and-place machine designs, the operator may refill feeder slots while the machine is processing PCBs (Carmon et al. 1989). For other machine designs, the number of

components loaded during a job setup is sufficient to complete a batch (1000-4000 components) (Ahmadi and Kouvelis 1994).

There are two elements to the manufacturing time, M_X, for batch $\boldsymbol{X}$. The setup time, s, is the time to prepare the machine to process a batch. The processing time, P_X, is the time it takes the machine to insert all components on all of the PCB jobs in the batch. The processing time is dependent on the component-type-to-feeder-slot assignment, the travel time from the feeder slots to the home position, and the batch profile, R^X. Creating the optimal component-type to feeder-slot assignment minimizes the processing time for a batch. As shown by Magazine and Polak (2003), the optimal component-type to feeder slot assignment is an organ-pipe configuration. The organ-pipe configuration orders the component types in the job profile by cardinality. This ordering of component types is then assigned to the feeder slots by increasing distance from the home position thereby allocating the most frequently occurring components in the batch to the feeder slots closest to the home position.

The processing time for batch $\boldsymbol{X}$, P_X, with batch profile R^X can be defined using:

1) $U_X^*(i)$ denotes the index of the feeder slot to which component i is assigned in the optimal organ pipe configuration for batch $\boldsymbol{X}$.
2) d_j denotes the time required to retrieve a component from feeder slot j and insert it into the PCB.

Then the processing time for batch $\boldsymbol{X}$ is $P_X = \sum_{i \in N} d_{U_X^*(i)} r_i^X$

and the manufacturing time for batch $\boldsymbol{X}$ is $M_X = s + P_X$ (Cohn et al. 2001).

Consider a batch $\boldsymbol{X}$ that contains jobs *a, b,* and *c,* and compare the processing time for the batch, P_X, to the total of the processing time for the individual jobs, $P_a+P_b+P_c$. We know from previous work that $P_x \geq P_a+P_b+P_c$ due to *superadditivity* (Magazine and Polak 2003). Intuitively, this increase in processing time occurs because the optimal component-type to feeder-slot assignment for the batch differs from that of the individual jobs. Since the assignment is no longer optimal for the individual jobs the processing time for job *a* using the batch setup is greater than P_a. This leads to a trade-off between setup time, which is reduced when we process jobs in a batch, and processing time, which usually increases when we process jobs in a batch.

In summary, the goal of the PCB-JB problem is to determine a partition of the set $\boldsymbol{K}$, such that the total manufacturing time (setup time plus processing time) is minimized for the entire set of jobs. All partitions of the set $\boldsymbol{K}$ are feasible solutions. Due to the operating and technology characteristics, the optimal partition that has the minimum manufacturing time is independent of the

sequence of the jobs within the batch. The optimal partition is also independent of the sequence of batches.

Crama et al (1999) have proposed a classification structure for the PCB planning process noting that these optimization problems are interdependent. As discussed by Cohn, Magazine, and Polak (2001) and Magazine and Polak (2003) the PCB-JB problem is an integration of two sub-problems of the classification SP1 and SP 5, product clustering and machine setup.

HEURISTIC FAMILIES

We briefly introduce the best variation from each of the four heuristic families. More detail on the development and testing of these heuristics and algorithm specifics are given in Williams and Magazine (forthcoming). For all four heuristics families, the best variation was the heuristic variation with the smallest average and variance for the percent difference from minimum.

Clustering Family Heuristic

Cluster analysis is a commonly used technique to create mutually exclusive, disjoint sets of similar objects. Anderberg (1973 p. 10) states that "The foremost difficulty is that cluster analysis is not a term for a single integrated technique with well defined rules or utilization; rather it is an umbrella term for a loose collection of heuristic procedures and diverse elements of applied statistics." There are many excellent references on cluster analysis (Anderberg 1973, Everitt 1993).

The variation selected in Williams and Magazine (forthcoming) utilizes a similarity measure and clustering algorithm tailored specifically to the PCB-JB problem. An outline of the clustering heuristic is given in Figure 2. The similarity measure is processing time difference which measures the similarity between two batches the difference between the processing time for the batches processed together and the processing time for the batches to be processed individually. The comparison variable used to compare batches by the similarity measure is the batch profile, R^x.

Bin-packing Family Heuristic

The bin-packing problem is defined as: Given a set of items, $\boldsymbol{L} = \{l_1, l_2, \ldots l_K\}$ with size $v(l_j)$ Î (0,1], find a partition of $\boldsymbol{L}$ with disjoint subsets $L_1, L_2, L_3, \ldots, L_P$ such that the sum of the sizes of the items in each L_p is less than or equal to 1 and the number of partitions, P, is minimized (Coffman et al. 1997). In the PCB-JB problem we have K jobs that are to be partitioned into an unknown number of batches (bins) so that manufacturing time is minimized. With an appropriate definition for a job 'fitting' into a batch, we expected the bin-packing approximation algorithms to perform well on the PCB-JB problem. We focused on the first-fit decreasing and best-fit decreasing bin-packing approximation algorithms. Recall that first-fit decreasing orders the list of items, $\boldsymbol{L}$, in non-increasing order. The packing rule for first-fit is to find the first (lowest index number) bin that has

capacity for the next item, l_i, in the list. If no bin has sufficient capacity, then a new bin is created and the next item is placed in the new bin. Best-fit decreasing operates the same as first-fit decreasing except that the packing rule is modified. The packing rule for best-fit decreasing is to find which bin has availability closest to $v(l_i)$ and place the next item, l_i in this bin (Coffman et al. 1997).

Figure 2: Clustering Family Heuristic

Best Cluster family heuristic:
PCBCluster Algorithm with Processing Time Difference Similarity Measure

- Initialize b=K batches with 1 job in each and determine total manufacturing time for this solution
- Calculate processing time difference similarity matrix
- Repeat
  - $b \leftarrow b$-1
  - Search similarity matrix for "most similar" pair
  - Merge the two batches that are "most similar" thereby creating a 'new' batch
  - Calculate total manufacturing time for this solution with b batches
  - Calculate new similarity matrix
- Until all jobs are in a single batch (K-1 times)
- K solutions have been determined. Select the solution with the lowest total manufacturing time.

An outline of the bin-packing heuristic is given in Figure 3. The primary issue in this heuristic is how to decide if a job 'fit' with one of the existing batches. Various definitions of 'fit' were tested and the best results were obtained using the definition that a job 'fit' if the increase in processing time did not exceed a percentage, p, of the setup time. Thus, if we are considering adding job i to batch $\boldsymbol{X}$, the test is P_{Xi}-P_X-P_i £ ps. The best value for p varied from problem to problem so a grid search on p is performed as part of the heuristic.

Genetic Algorithm Shortest Path Problem (GASPP) Family Heuristic

Sequencing genetic algorithms, also called order-based genetic algorithms, encode a solution as a permutation of integers in the range of 1..K. Such an encoding requires the use of reordering crossover operators. The sequencing genetic algorithm has been successfully applied to different types of sequencing problems such as the traveling salesman problem (TSP) (Chatterjee, Carrera and Lynch 1996), (Schmit and Amini, 1998), clustering (Cowgill, Harvey, and Watson 1999),

manufacturing cell design (Joines, Culbreth, and King 1996), and flow shop sequencing (Reeves 1995). See Reeves (1993) for a general overview and other applications.

In the GASPP family heuristic, we exploit a special case of the PCB-JB problem noted by Magazine and Polak (2003). Given a fixed sequence of jobs, a shortest-path algorithm can determine the optimal batching. The sequencing GA searches over the space of all job sequences. A shortest-path algorithm evaluates the fitness of each sequence by finding the optimal batching of the sequence.

Figure 3: Bin-packing family heuristic

```
Best Bin-packing family heuristic

Loop: For p = 1.00 down to p = 0.11 by 0.01
        Order jobs by decreasing job size
        Create batch 1 with first job in the ordered list
        Loop: For each job i in the ordered list
                Does job i fit with one of the existing batches? P_Xi-P_X-P_i ≤ pσ
                If yes then
                        Determine which batch is the best fit for this job, min[P_Xi-P_X-P_i]
                        Add job i to the best-fit batch
                        Determine profile of new batch
                If no then
                        Create a new batch and add job i to this batch
        End Loop
        Determine total manufacturing time for this solution
End Loop
Select solution that has the lowest total manufacturing time
```

The GASPP heuristic developed in Williams and Magazine (forthcoming) is listed in Table 1. This heuristic utilized a steady-state evolutionary strategy (Reeves, 1993). The initial population was created by randomly initializing a sequence for each individual. The parent selection was rank selection (Reeves, 1993). The edge recombination crossover operator Whitley et al. 1989) and reverse mutation operator (Michalewicz 1996) were utilized.

Grouping problems partition a set of objects into a collection of mutually disjoint subsets in order to optimize a cost function. Many types of grouping problems such as the bin-packing problem and the graph-coloring problem are NP-hard. Traditional genetic algorithms (GAs) have not been successful at solving these types of problems. Falkenauer (1998) discusses the drawbacks of previous GA implementations to solve grouping problems. Primary among these drawbacks is that traditional crossover operators work with the objects in a group rather than the groups. He proposes a new encoding scheme and operators that manipulate the groups instead of the objects in the groups and calls the resulting heuristic GGA (grouping genetic algorithm).

We developed a crossover operator and a mutation operator for a GGA to solve the PCB-JB problem. The elements of the GGA heuristic are listed in Table 2. We utilized a steady-state

evolutionary approach with binary tournament parent selection (Reeves 1993). We modified the basic GGA crossover operator that is outlined by Falkenauer (1998) by including a 1-opt procedure. In the 1-opt procedure, jobs from a merged batch are moved one at a time to each of the other batches to see if an improved total manufacturing time can be achieved. Despite our significant development effort, there are indications that additional development work would still be useful.

Grouping Genetic Algorithm (GGA) Family Heuristic

Table 1 Best GASPP family heuristic	
GA Configuration	GA Parameters
Population initialization	Population size
Random sequence	3*Number of jobs
Evolution strategy	Mutation rate
Steady-state evolution	0.15
Parent selection strategy	Termination criteria
Rank selection	Varied by problem size
Crossover operator	Number of replications
Edge recombination	Varied by problem size
Mutation operator	
Reverse	

Table 2: GGA family heuristic	
	Best GGA Family Heuristic
Evolution strategy:	Steady state
Selection strategy:	Binary tournament
Initial population:	Random sequence with optimized batches
Crossover:	Falkenauer's 5-step with 1-opt adaptation
Mutation:	1-opt on randomly selected group
Probability of Mutation:	0.3

COMPARISON STUDY

We used an experimental design to compare the performance of each of the four heuristics. Based on the testing in Williams and Magazine (forthcoming), we anticipated that none of the

heuristics would perform best under all circumstances. We wanted to identify which heuristic to use for specific problem characteristics.

We ran four experiments, one for each heuristic, and measured two response variables, solution quality and execution time of the heuristics. We selected three measures of solution quality to examine:

1) Percent difference from minimum is the percent difference of the heuristic solution from the minimum solution found. The minimum solution found is the minimum of the four heuristic solutions determined for any particular problem instance.
2) Percent difference from optimal is the percent difference of the heuristic solution from the optimal solution.
3) Rank is the rank of the heuristic solution compared to the other heuristic solutions. The best solution is given a rank of 1. If more than one heuristic finds the best solution, then both heuristics are assigned a rank of 1 and the remaining heuristics are ranked 2 and 3.

We selected 5 factors to investigate in this experiment. During heuristic development (Williams and Magazine, 2007), these factors were observed and appeared to affect the solution quality obtained by a heuristic.

- Problem size has two elements: the *number of jobs in the planning set* and the *number of component types*.
- Type of *board profile*.
- *Setup time* had been observed to significantly affect the execution time in the optimal algorithms developed by Cohn et al. (2001).
- *Job size variance* affects the job profile.

We used a resolution V, 2^{5-1} fractional factorial design. This design allowed us to examine the main effects without any confounding. The two-way interaction effects would only be confounded by three-way and higher interactions (which we assumed were negligible). The 2^{5-1} fractional factorial design contains 16 design points (DP). We replicated the design 5 times. Therefore a total of 80 problems were randomly generated based on parameters from the literature. Of these 80 problems, 38 could be solved to optimality using the algorithm developed by Cohn et al. (2001).

We utilized a data set and information from the literature (Ahmadi and Kouvelis, 1999), (Jain, Johnson, and Safai, 1996), (Leon and Peters 1998) to determine the high/low values for the factors (Table 3). We obtained a data set from Marco Perona of the Politechnico di Milano. This data set is from ITALTEL and is described in Luzzatto and Perona (1993) and contains the bill-of-materials for 126 different board types which requires 250 component types. This data set was primarily used to determine PCB board profile characteristics.

Table 3: High/Low Factor Values for Experiment					
	Number of jobs	Number of Components Types	Setup Time	Job Size Variance	Profile
	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Low Value (-)	32	15	Med	Lo	Flat
Hi Value (+)	100	60	Hi	Hi	60::20

Number Of Jobs

The range in the literature for the number of jobs is as low as 5 and as high as 100. The IP approach developed by Cohn et al. (2001) can solve to optimality most problems with up to 32 jobs. For the low value of the number of jobs factor we selected 32. This allowed us to find optimal solutions for most of the small problems.

Number Of Component Types

In the ITALTEL data set and the literature, boards typically have a maximum of 60 component types. We used 15 for the low value and 60 for the high value.

Setup Time

Setup time significantly affects the number of batches in the optimal partition. For the low value of the factor, we estimated a setup time that would produce an optimal partition with the number of batches near the median ([K+1]/2). Our experience and that of Cohn et al. (2001) indicated that problems that have the optimal number of partitions near the median are more difficult to solve using integer programming (IP).

The high/low values of setup time were determined using this procedure: We first used an enumeration procedure to determine an estimate of *MaxSetupK*, the largest that a small setup time can be and still have K batches in the optimal partition, and *MinSetup1,* the smallest that a large setup time can be and still have all jobs in a single batch in the optimal partition. We define *Range* = *MinSetup1* – *MaxSetupK* and *Mid* = (*MinSetup1* + *MaxSetupK)*/2. Then for a 'Med' value of setup time we randomly sampled from the distribution U(*Mid*-0.05*Range*, *Mid* + 0.05*Range*). For a 'Hi' value of setup time we sampled from the distribution U(*Mid*+0.3*Range*, *Mid*+0.4*Range*).

Job Size Variance

We generated job sizes within the range of the job sizes found in the literature (Ball and Magazine 1988, Jain et al. 1996, Leon and Peters 1998). The variance of the jobs sizes in a planning

set was the factor that we manipulated in these experiments. For the low value of the factor, the job sizes in a planning set were uniformly distributed between 50 and 60 with a variance of 10. For the high value of the factor, the jobs sizes in a planning set were uniformly distributed between 10 and 100 with a variance of 690. For both, the expected job size is 55.

Board Profile

There is little information in the literature regarding board profiles. We utilized the ITALTEL data set to examine board profiles more closely. The average percentage of placements for the top 20% component types is approximately 60%. Therefore, one board profile has the top 20% of the component types containing 60% of the placements and is referred to as the 60::20 profile.

The second board profile is a flat profile. In a flat profile, the number of each component type is uniformly distributed. The parameters of the distribution were selected so that the number of placements on a PCB is typical of values found in the literature.

SOLUTION QUALITY ANALYSIS

For each heuristic, we performed four different analyses with regard to the response variable solution quality.

1) Multiple means comparison using the Least Significant Difference (LSD) procedure performed by SAS Version 6.12 for Windows. The design points (DPs) are divided into groups such that the mean percent difference from minimum is not significantly different. Then DPs are examined for common problem characteristics.
2) Comparison with optimal solutions (only problems with a small number of jobs). We examined the characteristics of the design points at which each heuristic finds the optimal solution.
3) Main and interaction effects were determined for the five factors of the experimental design.
4) In a response surface analysis, we estimated a regression equation that predicts the solution quality of a particular heuristic given problem characteristics number of jobs, number of component types, setup time, job size variance, and profile. We regressed on the five factors and any significant interaction effects found in the main effects and interaction analysis (See (3) above).

The clustering heuristic

- had three significant factors: number of jobs, number of components, and board profile.

- did not find any optimal solutions for the small problems.
- on average, was 0.81% away from the optimal with a standard deviation of 0.55% for the small problems.

The bin-packing heuristic

- had three factors with a significant effect on solution quality: board profile, job size variance, and the number of jobs.
- did not find any optimal solutions for the small problems.
- on average, was 1.03% away from the optimal with a standard deviation of 0.68% for the small problems.

The GASPP heuristic

- had four factors with a significant effect on solution quality: number of jobs, number of component types, board profile, and job size variance.
- had three significant interactions: 1) The largest interaction effect was between the number of jobs and the number of component types. The magnitude of this effect was close to the magnitudes of the main effect. 2) There was a significant interaction between the number of jobs and board profile. 3) There was a significant interaction between the number of component types and the board profile.
- found the optimal solution for 4 of the 38 small problems.
- on average, was 0.12% away from optimal with a standard deviation of 0.14% for the small problems.

The GGA heuristic

- had four factors with a significant effect on solution quality: number of jobs, board profile, job size variance, and number of component types.
- found optimal solutions for 27 of the 38 small problems. Of the 11 problems where the optimal solution was not found, 10 terminated because the population had converged before the specified number of new individuals were created. This demonstrates that improvements can be made to the GGA.

EXECUTION TIME ANALYSIS

We measured execution time using the timing routines in Borland's Version 4.5 of C/C++. All problems for all heuristics were executed on a 600 megahertz machine with 128 megabytes of RAM and Windows 98. Input processing was excluded from the execution time.

A summary of execution time is given in Table 4. Clustering and the bin-packing heuristic are the fastest algorithms. Clustering and the bin-packing heuristic are both heuristics that construct a solution by passing through the list of jobs one time. Clustering, on average, is about ½ second faster than the bin-packing heuristic. The bin-packing heuristic is slightly slower than clustering because it performs a grid search on one of its parameters.

Table 4: Execution time comparison for all heuristics

	Execution Time (secs) for Small Problems		Execution Time (secs) for Large Problems		Average Execution Time (secs) for All Problems	Maximum Execution Time (secs) for All Problems
Heuristic	Average	Std. Dev.	Average	Std.Dev		
Clustering	0.4	0.3	11.0	7.2	5.7	17.8
BFDJB	1.5	1.3	10.8	10.0	6.2	27.5
GASPP	886.1	583.3	33660.3	21821.3	17273.2	54332.0
GGA	662.2	505.0	44172.7	39066.8	22417.5	119490.0

GASPP and GGA search a solution space. GASPP searches the space of all sequences of the jobs and GGA searches the space of all possible partitions of the set of jobs. Thus, as the problem size increases the execution time increases dramatically. GGA outperforms GASPP on the small problems by and average of 3.7 minutes (224 seconds). However, on average due to the large problems, GASPP outperforms GGA by 85.7 minutes (5144.3 seconds).

Compared to the optimal algorithm of Cohn, Magazine and Polak (2001) the GGA is faster for 26 of the 38 small problems. The average execution time using the optimal algorithm for 38 of the 40 small problems was 380 minutes with a standard deviation of 847 minutes. For the same 38 small problems, execution time for GGA was 11 minutes and the standard deviation was 8.7 minutes.

RESULTS

There were 38 small planning set problems for which the optimal solution could be obtained. All four heuristics found solutions that are within 3% of optimal (Table 5). For the large planning set problems, each heuristic solution was compared to the best solution (minimum solution). The maximum percent difference from the minimum is 4.67%. There is not a single best heuristic that achieves the best solution quality for all types of problems. Clustering finds the minimum solution in 24 of the 80 problems (Table 6) and GGA finds the minimum solution in 37 of the 80 problems. GASPP and the bin-packing heuristic are most frequently ranked 3rd.

Table 5: Solution Quality Summary for All Heuristics: Percent Difference from Minimum and Optimum						
	Percent Difference from Minimum (80 problems)			Percent Difference from Optimal (38 problems)		
Heuristic	Average	Std. Dev.	Maximum	Average	Std.Dev	Maximum
Clustering	0.49%	0.56%	2.37%	0.81%	0.55%	2.37%
Bin-packing	0.84%	0.79%	3.21%	1.03%	0.68%	2.78%
GASPP	0.56%	0.67%	2.97%	0.12%	0.14%	0.72%
GGA	1.07%	1.39%	4.67%	0.01%	0.03%	0.19%

Table 6: Solution Quality Summary for All Heuristics: Ranking Information				
	Ranking (80 problems)		Number of times found worst solution	Number of times found best solution
Heuristic	Mode	Frequency		
Clustering	1	24	16	24
BFDJB	3	31	18	16
GASPP	3	36	4	9
GGA	1	37	36	37

Of the five problem characteristics that we tested, setup time was the only factor that had no significant effect on either solution quality or execution time for any of the heuristics. This is surprising and interesting given the impact that setup time has on the execution time of the optimal algorithms. Board profile was the only problem characteristic that affected solution quality performance in the same manner across all heuristics. The quality of solutions found for problems with a flat board profile were higher than for problems with a 60::20 board profile. The effect on solution quality and execution time of the other three problem characteristics, number of jobs, number of components types, and job size variance, was dependent on which heuristic was used.

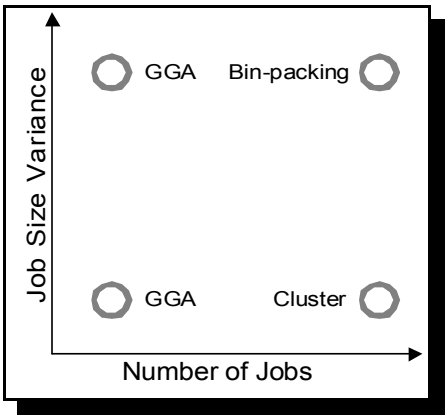
We select the best heuristic to use based on the problem characteristics (Figure 4).

- The clustering heuristic found the best solutions for problems with a large number of jobs and small job size variance. Overall, clustering had the best average solution quality and was the fastest algorithm.
- Overall, bin-packing had the third best average solution quality. The bin-packing heuristic is best for problems with a large number of jobs and a large job size

variance. This heuristic was on average 0.5 seconds slower than the clustering heuristic.

- The GASPP heuristic never finds the best solution. However, on average it is second best. This heuristic was much slower than the clustering heuristic requiring an average of 4.8 hours. Despite its good average performance, GASPP does not outperform the other heuristics for any of the studied problem characteristics.
- GGA found the best solutions for problems with a small number of jobs. In fact, it found optimal solutions for 27 of the 38 problems with known optimal solutions. However, the average solution quality for GGA is the worst of all four heuristics. This heuristic is the slowest on average requiring an average execution time of 6.2 hours. The essential failure of the GGA heuristic for large problems along with our experience in the development phase indicate that despite our significant development effort, additional improvement in the GGA heuristic is still needed. The success of GGA on small problems indicates that GGA has a potential that that should be explored in further work.

Figure 4: Significant factors for selecting a heuristic



One might conclude that good solutions are easy to find. However, the results from the heuristic variations that were tested during heuristic development do not support such a statement (Williams and Magazine, forthcoming). In particular, there were variations that were not selected for the experiment that were on average 2-5% from the best solution found and as much as 15% from the best solution. Hence, a careful choice of heuristic is important.

Execution time varies significantly among the heuristics. The clustering and bin-packing algorithms are fast enough to be considered on-line algorithms. Clustering is the fastest. GASPP

and GGA are search heuristics and are off-line algorithms. Execution time could be the determining factor in selecting the heuristic of choice.

If the experiment were to be re-run on more up to date computer hardware we would expect both GASPP and GGA to find higher quality solutions for the same amount of execution time. However, we believe the comparative performance of the four heuristics on the various problem characteristics would not be significantly different.

CONCLUSIONS AND RECOMMENDATIONS

We have compared the performance of four heuristics on the PCB-JB problem using an experimental design. There is yet some untapped potential in the GGA heuristic. Early convergence for some of the small problems indicates that some modification might improve results. In addition, though it finds more optimal solutions than the other heuristics it finds the lowest quality solutions for large problems. Additional work to reduce the execution time could improve solution quality for the large problems.

All four heuristics find high-quality solutions. Based on this study, we make the following recommendations:

1) If a single heuristic must be selected to solve all types of PCB-JB problems, use the clustering heuristic. It achieves the best average solution quality and executes the fastest.
2) If the problem characteristics are known and execution time is not crucial:
   - ♦ For problems with a small number of jobs, use the GGA heuristic.
   - ♦ For problems with a large number of jobs and small job size variance, use the clustering heuristic.
   - ♦ For problems with a large number of jobs and large job size variance, use the bin-packing heuristic.
3) If the problem characteristics are known and an on-line algorithm is required:
   - ♦ For problems with small job size variance, use the clustering heuristic.
   - ♦ For problems with large job size variance, use the bin-packing heuristic.

REFERENCES

Ahmadi, J., S. Grotzinger & D. Johnson. (1988). Component Allocation and Partitioning for a Dual Delivery Placement Machine. *Operations Research,* 36, 176-191.

Ahmadi, R.H. & P. Kouvelis. (1994). Staging Problem of a Dual Delivery Pick-and-Place Machine in Printed Circuit Card Assembly, *Operations Research,* 42, 81-91.

Ahmadi, R.H. & P. Kouvelis, (1999). Design of Electronic Assembly Lines: An Analytical Framework and Its Application, *European Journal of Operational Research*, 115, 113-137.

Anderberg, M.R. (1973). *Cluster Analysis for Applications*. New York: Academic Press.

Ball, M. O.& M.J. Magazine. (1988). Sequencing of Insertions in Printed Circuit Board Assembly. *Operations Research,* 36, 192-201.

Carmon T.F., O.Z. Maimon & E.M. Dar-Els. (1989). Group Set-up for Printed Circuit Board Assembly. *Int. J. Prod. Res.,* 27, 1795-1810.

Chatterjee, S., C. Carrera & L.A. Lynch. (1996). Genetic algorithms and traveling salesman problems. *European Journal of Operational Research,* 93, 490-510.

Coffman, E.G., M.R, Garey & D.S. Johnson. (1997). Approximation Algorithms for Bin Packing: A Survey. D. Hochbaum, ed. *Approximation Algorithms for NP-Hard Problems.* Boston: PWS Publishing Co.

Cohn, A.M., M.J. Magazine & G.G. Polak. (2001). Integrating Printed Circuit Board Manufacturing Problems by Branch-and-Price. ORC Report OR 01-348, Cambridge, MA: MIT.

Cowgill, M.C., R.J. Harvey & L.T. Watson. (1999). A Genetic Algorithm Approach to Cluster Analysis. *Computers and Mathematics,* 37, 99-108.

Crama Y, van de Klundert J & Spieksma FCR. (2002). Production planning problems in printed circuit board assembly. *Discrete Applied Mathematics,* 123, 339-361.

Depuy, G.W., M.W.P.Savelsbergh, J.C. Ammons & L.F.McGinnis. (2001). An Integer Programming Heuristic for Component Allocation in Printed Circuit Card Assembly Systems. *Journal of Heuristics,* 7, 351-369.

Everitt, B.S. (1993). *Cluster Analysis*. London: Edward Arnold.

Falkenauer, E. (1998). *Genetic Algorithms and Grouping Problems*. Chichester, England: Wiley & Sons

Jain, S., M.E. Johnson & F. Safai. (1996). Implementing Setup Optimization on the Shop Floor. *Operations Research,* 43, 843-851.

Joines, J.A., C.T. Culbreth & R.E. King. (1996). Manufacturing Cell Design: An Integer Programming Model Employing Genetic Algorithms. *IIE Transactions,* 28, 69-85.

Leon, V.J. & B.A. Peters. (1998). A Comparison of Setup Strategies for Printed Circuit Board Assembly. *Computers & Ind. Eng.,* 34, 219-234.

Luzzatto, D., M. Perona. (1993). Cell Formation in PCB Assembly Based on Production Quantitative Data. *European J. of Operational Research,* 69, 312-329.

Magazine, M.J. & G.G. Polak. (2003). Job Release Policy and Printed Circuit Board Assembly. *IIE Transactions,* 35, 379-388.

Magazine, M.J., G.G.Polak & D. Sharma. (2002). A Multi-Exchange Neighborhood Search Heuristic for an Integrated Clustering and Machine Setup Model for PCB Manufacturing. *J. of Electronics Manufacturing*., 11, 107-119.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin: Springer-Verlag.

Mody, A., R. Suri & M. Tatikonda. (1992). Keeping pace with change: international competition in printed circuit board assembly. *Industrial and Corporate Change,* 4, 583-614.

Nakahara, H. (1999). PCB Output 1998. *Printed Circuit Fabrication,* 22, 38-40.

Nakahara, H. (2001). The Top PCB Shops. *Printed Circuit Fabrication,* 24, 26-31.

Nakahara, H. (2003). China: More capacity but at what cost? *Printed Circuit Fabrication,* 26, 32-37.

Reeves, C.R. (1993). Genetic Algorithms. C.R. Reeves, ed. *Modern Heuristic Techniques for Combinatorial Problems*. New York: Halsted Press, 151-196.

Reeves, C.R. (1995). A Genetic Algorithm for Flowshop Sequencing. *Computers Ops. Res.,* 22, 5-13.

Schmitt, L.J. & M.M. Amini. (1998). Performance Characteristics of Alternative Genetic Algorithmic Approaches to the Traveling Salesman Problem Using Path Representation: An Empirical Study. *European Journal of Operational Research,* 108, 551-570.

Whitley, D., T. Starkweather, D. Fuquay. (1989). Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator. J.D. Schaffer, ed. *Proc. 3rd Int'l Conf. On Genetic Algorithms.* Los Altos, CA: Morgan Kaufman Publishers, 133-140.

Williams, S.K. & M.J. Magazine. (2007). Heuristic approaches to batching jobs in printed circuit board assembly. *Computers and Operations Research,* 34, 1943-1962.

THE INFLUENCE OF DECISION COMMITMENT AND DECISION GUIDANCE ON DIRECTING DECISION AID RECOMMENDATIONS

Darryl J. Woolley, University of Idaho

ABSTRACT

This study considers factors that influence whether decision-aid users direct decision aid recommendations toward their prior beliefs. Desiring to confirm their beliefs, decision aid users may seek to direct decision aid recommendations toward their prior opinions. Cognitive dissonance theory suggests that this is more likely to occur when users are strongly committed to their opinions. However, if decision-makers receive guidance from a decision aid, they may be less likely to direct decision aid recommendations toward their prior belief. In the first of two experiments, professional auditors were more likely to direct decision aid recommendations if they were committed to a decision before using the aid. In the second experiment, graduate business students were more likely to accept a decision aid's initial recommendation when the decision aid provided guidance. However, the decision aid guidance did not stop users from directing the decision aid recommendation toward their prior belief; rather, it appeared to influence users who otherwise would have shifted away from both the decision aid recommendation and their own prior belief. This study contributes to research on decision aid use by finding that both professional and non-professional decision aid users direct decision aid recommendations toward their prior belief, and that they are influenced by the degree of their decision commitment and the guidance they receive from the decision aid.

INTRODUCTION

In March 1942, the Japanese naval staff began to plan a campaign designed to draw the United States navy into a decisive battle near the Pacific island of Midway. The campaign had been proposed by Admiral Matome Ugaki in January. As part of the planning, the naval staff conducted a simulation to determine the feasibility of the plan. Admiral Ugaki refereed the simulation. As a strong supporter of the campaign, "… he allowed nothing to happen which would seriously inconvenience the smooth development of the war games to their predestined conclusion. He did not scruple to override unfavorable rulings of other umpires." (Prange, et. al., 1982, p. 31)

The battle simulation used to plan the battle of Midway is an example of using a decision aid, a tool used to assist its user in improving a decision. Decision aids are used in a variety of contexts, including medical diagnosis, bankruptcy prediction, and audit tasks. Research on decision

aids has investigated whether they yield superior judgments (Dawes, Faust and Meehl 1989) and, especially in the accounting domain, whether and why decision makers decide whether to rely and decision aid decision recommendations.

As Admiral Ugaki's war game suggests, a user may face more choices than to rely or not on an aids recommendation. A user may control the aid's result so that it is consistent with his or her own opinion. Research in accounting suggests that auditors control a decision aid results to concur with their personal opinion (Kachelmeier and Messier, 1990; Messier, Kachelmeier and Jensen, 2001). Organizations implement decision aids to guide or direct a decision process (Silver, 1990); understanding why people control decision aid recommendations and how this behavior effects decisions will help an organization match its decision support technology to its strategy. For an aid that is designed to guide decisions to add value to an organization, the aid should produce a different decision than an unaided judgment. By controlling a decision aid's recommendation to confirm a prior belief, users obtain the same decision as they would without use of the aid and diminish the aid's value. Decision aids are expensive to develop and often fall into disuse (Gill 1995). Whenever an organization implements a decision aid, it faces the threat that the aid's users will circumvent the aid's recommendation to rely on their own judgment.

This paper builds upon Kachelmeier and Messier (1990) to investigate why people control decision aid results and the effectiveness of strategies to reduce decision aid controlling. An experiment with CPA participants tests whether decision commitment, a construct drawn from social psychology (Festinger, 1957), increases how much users control decision aid recommendations, and an experiment with MBA student participants tests whether using decision aid explanations reduces the extent that users control decision aid recommendations. The results partially support that decision commitment influences controlling decision aid recommendations and support that decision aid explanations reduces controlling decision aid recommendations.

THEORY AND HYPOTHESES

Decision aids usually improve decision results (Dawes, Faust and Meehl, 1989), but decision makers often choose to rely upon their own judgment rather than upon an aid's recommendation. Incentives, feedback, and justification requirements decrease reliance on decision aids (Ashton, 1990). Experienced or confident decision makers believe that their ability is adequate and are less likely to rely upon decision aids than less experienced users (Arkes, Dawes and Christensen, 1986; Whitecotton, 1996). Decision makers who are informed about the decision aid algorithm or who are able to interact with the aid have greater confidence in the aid and feel more in control of the decision process, and are more likely to rely upon the aid (Davis, 1998; Eining, Jones and Loebbecke, 1997).

The decision aid reliance research cited above examines decision aids with inputs provided by the researcher and measures reliance as the difference between decision aid recommendations and decision aid users' decisions. In comparing human judgment to decision aid effectiveness,

however, people are found to be better at measuring variables whereas decision aids are better at combining those inputs into a combined judgment (Klienmuntz, 1990). In practice decision aids often consist of subjective inputs based on users' judgment that are combined by the aid to arrive at a recommended coursed of action (e.g., Shelton, Whittington and Landsittel, 2001). Allowing users to decide the inputs of a decision aid may substantially alter how a decision aid is used. When the aid inputs are set externally to the user, the user is faced with either accepting the aid's recommendation or not accepting the recommendation. An aid that allows users to decide the inputs allows the user to use a decision aid recommendation to confirm their prior belief or to reach a strategic recommendation by setting the inputs.

Researchers in social psychology have long noted that people manage evidence based on their prior beliefs or social pressure (Festinger, 1957). Theories developed as a result of these observations provides a framework that describes a lack of reliance on decision aids, including managing an aid to get an expected result. These theories predict that people will use evidence to confirm beliefs to which they are strongly committed or to which they feel social pressure to conform. People search for or interpret evidence to support committed beliefs, opinions, or decisions, rather than rationally combine the evidence in the decision or belief-formulation process. If we regard a decision aid's recommendation as a piece of evidence for a decision-maker to consider, then we can investigate how the recommendation is treated as any other type of evidence is investigated.

The theory of cognitive dissonance proposes that a disagreement between beliefs or between a decision and new evidence may cause psychological (Elliot and Devine, 1994) or possibly physical discomfort (Kiesler and Pallak, 1976). Impression management theory predicts that people desire to appear consistent because of the others' expectations. Both theory predict that the more committed a person is to a decision, either internally or externally, the more they will manage evidence to agree with their prior belief. They create a balance between their former belief and new evidence by interpreting new evidence in a manner consistent with their former belief. "Rational" decisions based on an unbiased evaluation of all evidence could cause discomfort by forcing decision makers to become aware of their incorrect judgment or beliefs. Whether people are willing to change their beliefs based on new evidence depends upon the relative strength of their commitment to their prior belief and of the persuasiveness of the new evidence (Aronson, 1968). If decision makers are strongly committed to their prior belief, the path of least resistance to cope with disconfirming evidence is to avoid or to discredit the new information. For example, they may seek out evidence consistent with their prior belief or avoid disconfirming evidence (Frey, 1986). If new information cannot be avoided, people discredit the new information (Lord, Ross, and Lepper, 1979).

Commitment occurs when a decision is irreversible (Wicklund et al., 1976), publicly communicated (Jecker, 1963), linked to a subsequent action (Kiesler and Sakumura, 1966), or based on a strong belief (Brock et al., 1967; Sweeney and Gruber, 1984). If decision makers are not committed to their prior decision or belief, they simply reverse their decision or belief when

confronted with contrary information. For example, inexperienced decision makers over-rely on decision aids, using them when not appropriate because they have no committed opinion (Glover, Prawitt, and Spilker, 1997). People who are strongly committed to their decision defend their decision and are biased in their information evaluation or in their future decisions (Staw, 1976). Social psychology research usually manipulates decision commitment as a proxy for the presence of discomfort caused by inconsistent new information (Brehm and Cohen, 1962).

These findings also apply in professional contexts. Auditors, for example, indicate that the ability to justify or document a decision is more important than decision consensus, compromise, or finding a single best answer (Gibbins and Emby, 1984). They often make judgments early in the decision process and then gather evidence to support that judgment. They also point out the importance of protecting themselves from potential negative consequences of their decisions and that they often use audit working papers to justify decisions rather than to record audit procedures (Gibbins, 1984).

Kachelmeir and Messier (1990) and Messier, Kachelmeier and Jensen (2001) found an apparent example of auditors evaluating evidence based on existing opinion in an audit sample size task. They based their research on the AICPA's suggested sample size selection aid (Audit Sampling, 1983, 1999), and found that auditors select smaller sample sizes than recommended by the aid. When using the AICPA's decision aid, auditors make audit judgments concerning audit risk, tolerable misstatement, and the role of other audit procedures. The decision aid combines these subjective judgments to calculate a recommended sample size. They compared sample sizes obtained using the aid from two groups of participants. The first sample calculated a sample size by inputting audit judgments to the aid, but the second sample only supplied audit judgments without using the aid; the authors then calculated sample size by inputting the judgments into the aid. They found that participants that used the aid calculated a lower sample size than the aid calculated for participants who only provided the audit judgments. Apparently the auditors that participated in their study adjusted their audit judgments to obtain an acceptable sample size.

Auditors routinely make sample size decisions, and may have boundaries on what they believe are reasonable sample sizes. Strong belief is a form of commitment, and if Kachelmeier and Messier's (1990) auditors found that the sample size recommended by the decision aid was outside of their accepted range, they may have attempted to find a way to reconcile the aid to their opinion by controlling the aids recommendation through changing their audit judgments. The effect of decision commitment can be tested by changing the level of commitment. One way to manipulate commitment is to require people to communicate their belief before asking for a decision.

H1 When users' prior opinions differ from a decision aid recommendation, users committed to a decision before using a decision aid

A: will direct the decision aid recommendation toward a sample size consistent with their prior belief more often than other auditors and

B: will indicate less agreement with the decision aid than other auditors.

Various factors may influence the level of commitment to a belief. More experienced auditors are expected to have stronger beliefs about the range of appropriate sample sizes and to be more committed to their prior belief. More experienced decision-makers tend to be more confident, and confidence has been found to be associated with lower decision aid reliance (Whitecotton, 1996). Experienced and more confident auditors are therefore expected to react more strongly to the decision aid recommendation. Their reaction may consist of both directing the decision aid toward a lower sample size more than their less experienced or confident colleagues and of evaluating the decision aid more critically.

H2: More experienced and more confident users are expected to
A: direct the decision aid toward a sample size more consistent with their prior belief more than less experienced auditors, and
B: indicate less agreement with the decision aid than less experienced auditors.

We suppose that people act to maintain balance between new evidence and their prior beliefs. Thus, the participants in Kachelmeir and Messier (1990) control the decision aid recommendation to agree with their belief. Balance can also be reached if new evidence is strong enough to convince a person to change their belief to be consistent with the new evidence. A decision aid may include characteristics to increase user confidence in the aid's recommendation; by doing so, the aid design may lower how much users' control decision aid recommendations by adjusting inputs. A suggested decision aid characteristic to increase user reliance in a decision aid is guiding explanations. Just as a user guides a decision aid's recommendations, a decision aid may guide the user in the decision making process by suggesting decision steps and explaining judgments (Silver 1990). Decision aid guidance has been effective in encouraging information processing strategies (Todd and Benbasat, 1994), and when this guidance consists of an argument in favor of a decision aid recommendation decision aid users are more likely to adopt that recommendation (Eining, Jones, and Loebbecke, 1997).

Silver (1990) suggests that guidance may consist either of promoting a course of action (active understanding) or of supplying information about the decision process (passive understanding), and may be used to direct the decision process toward a desired conclusion, through a desired decision strategy, or to a more effective or efficient decision. This is especially relevant to auditing practice where decision consistency is a desired decision characteristic (Ashton, 1974). Explanation of decision aid recommendations has been found to increase agreement with a decision aid's outcome (Eining et al., 1997) and to increase belief change in the direction of the decision aid recommendation (Ye and Johnson, 1995). Simply informing users that not using an aid would result in poor performance increases reliance in an aid (Arkes et al., 1986), but providing feedback on an aid's underperformance decreases reliance (Kaplan, Reneau, and Whitecotton, 2000). Information on an aid's rule that appears invalid users decreases reliance (Ashton, 1990).

System guidance may consist of a listing of the rules by which a decision aid reaches its conclusions or a justification of the aid's conclusions (Ye and Johnson, 1995). The rule listing, sometimes known as a trace, originally developed as a list of rules applied in an expert system session, informs the user of the processes used by the decision aid to reach its conclusion. For example, a decision aid based on a formula may let the users know the formula's structure and weights. A rule listing promotes reliance in a decision aid if the rules appear valid or inspire confidence in the aid's user, but does not argue in favor of the decision aid's recommendation. It is informative rather than persuasive. If the rule listing increases the aid's validity in the user's perception, it increases the effectiveness of the decision aid's recommendation in influencing the decision of the user, and may reduce dissonance by persuading the user that the aid's method produces an accurate or acceptable recommendation. However, if the rule listing does not increase the aid's validity in the user's perception, it may reduce the user's reliance on the aid and increase directing the aid's recommendation.

Justification guidance, in contrast to a rule listing, attempts to persuade users that the aid's conclusion is valid. It consists of an explanation of why the rule applies, and therefore of why the explanation is valid. For example, a formula-based decision aid may explain the reason behind the formula's weights or why the formula itself is useful. Because it actively promotes the aid's conclusion, it should be more persuasive than a rule listing. Again, justification guidance may reduce the user direction of decision aid guidance by increasing the user's perception of the aid's validity.

Whereas decision commitment increases the strength of prior beliefs, decision aid guidance strengthens the new evidence presented by the decision aid recommendation, giving it more weight to overcome the user's prior belief and persuade the user to adopt the decision aid's unbiased recommendation.

H3: Participants who are guided when receiving a decision aid's recommendation will

A: direct the decision aid's recommendation toward the decision aid's initial recommendation more often than unguided participants

B: direct the decision aid recommendation toward their own prior decision less often than unguided participants.

These three hypotheses were tested in two experiments. The first experiment tested whether belief commitment increases the amount of directing decision aid results; if so, then the users likely use the aids to confirm prior beliefs. The second experiment tested whether structuring the decision aid to guide the users would reduce the amount of directing decision aid results.

EXPERIMENT 1

Experiment 1 tested whether auditors direct decision aid outcomes to confirm decisions and tests hypotheses one through three. Thirty-three auditors from several Big 5 firms participated in the experiment. The experiment was conducted on a Web page for the convenience of the participants and their employers. The average age of participants was 27, with an average of 31 months audit experience. Experience (t = 1.06, p = .16) and age (t = .393, p = .91) did not differ across experimental conditions.

The participants completed the same task as those in Kachelmeier and Messier (1990), except that the decision aid was computerized to enable data collection (Cook and Swain, 1993). Pilot testing found that the decision aid provides a larger sample size than auditor sample size judgments without the aid. The task consisted of reviewing a case and using a decision aid to indicate a sample size for substantive audit tests. Experimental steps are shown in Table 1. The participants used the decision aid after reviewing the instructions and reading the case.

Table 1: Experiment 1 Steps by Experimental Group

Step	Control Group	Decision-first Group
Step 1	Instructions	Instructions
Step 2	Case	Case
Step3		Decision
Step 4	Decision aid	Decision aid
Step 5	Final questions	Final questions

To use the decision aid, the participants input three subjective judgments: tolerable misstatement, an assessment of combined control and inherent risk, and a reliance on other audit procedures. Tolerable misstatement is "a planning concept [that is] related to the auditor's preliminary judgments about materiality levels in such a way that tolerable misstatement, combined for the entire audit plan, does not exceed those estimates" (Audit Sampling, 1983). It is the dollar amount of misstatement the subject judges can be present before the financial statements are not reasonably presented. To enter the risk assessment and reliance on other audit procedures, the participants selected from a menu of items ranging, for the risk assessment, from Low to Maximum, and for the reliance on other procedures, from None to Substantial. Both menus consisted of four items. After entering the inputs the aid users clicked on a button to compute the sample size. They could repeat the process as often as they wished. They clicked on a finish button to signal that they were finished with the decision aid. When they clicked on the finish button, they were asked if they

had finished calculating the sample size. After using the decision aid, the participants completed a questionnaire that gathered demographic information and information about their satisfaction with the aid.

Kachelmeier and Messier (1990) compared sample sizes calculated from participants' audit judgments in the control group and sample sizes from participants' use of the decision aid in the experimental group. The average sample size from the experimental group was significantly lower than in the control group. To test if commitment is responsible for lowering the calculated sample size, we break Kachelmeier and Messier's experimental group into two different groups and vary the commitment. The participating auditors were randomly assigned into one of the two groups. Our control group is equivalent to Kachelmeier and Messier's experimental group. The final decision aid recommendation served as the control group's sample size decision. In Kachelmeier and Messier, auditors in this condition directed decision aid outcomes to a result that were, on average, between auditor's unaided sample size judgment and a sample size derived from auditor judgments of decision aid parameters, input into the decision aid by the researchers without the opportunity for decision aid user manipulation of results.

The participants in the second group, hereafter called the decision-first group, provided an unaided sample size after reading the case but before using the decision aid. They then used the sample size decision aid. Because they communicated their decision before using the aid, auditors in the decision-first group were expected to be more committed to a lower sample size than those in the control group, and therefore to direct the decision aid recommendation to a lower value.

Unlike Kachelmeier and Messier (1990), who required their participants to calculate the recommended sample size using a calculator, we were able to gather the participants' sample size results from first to last calculation through a computerized decision aid. They were able to gather only the participants' final decision. Because we gathered the decision aid recommendation from the first time the participants put their audit judgments into the aid and through every time they changed those judgments to obtain a different sample size, we could measure the change from the first to the last use of the ad. We found that participants often used the aid to recalculate a sample size after first use. The subjects iterated through the decision aid an average of 4.2 times, with a minimum of 1 and a maximum of 20. The number of iterations did not differ significantly between experimental groups. On average, aid users in both groups decreased the sample size from their first iteration to their last iteration using the decision aid. The first time they used the aid, the decision aid's recommended mean sample size was 391; but the last decision aid iteration produced an average recommendation of 65. The Wilcoxan Signed Ranks test, used because of non-normality, was significant, $Z = 2.782$, $p < .01$. With 2 outliers removed, the test was still significant, $Z = 2.396$, $p < .05$. Results did not qualitatively differ in any other test between the full sample and the sample with outliers removed. Therefore, the participants in this task performed in a similar manner to the experimental groups in the prior research (Kachelmeier and Messier, 1990; Messier, Kachelmeier, and Jensen, 2001).

Because the participants could use the decision aid multiple times to recalculate a sample size, we were able to measure both the aid's first recommendation and last recommendation. Hypothesis 1A was investigated with three tests: the decision aid sample size recommendation means, the ratio of the first use decision aid recommendation to the last use decision aid recommendation, and the proportion of participants that change their inputs to the aid so that the aid recommendation becomes a more acceptable value. If the decision-first group were to direct the decision-aid toward their original belief more than the control group, the decision-first group should arrive at a lower final sample size. The means of both the first and last recommendations do not differ between the control and decision-first groups, however (Table 2). To further test Hypothesis 1A, the ratio of the initial decision aid iteration to the last decision aid iteration was compared between groups. The decision-first group was expected to decrease the recommended sample size more than the control group through use of the decision aid and therefore to have a larger ratio of initial recommendation to last recommendation. As shown in Table 2, the ratio is higher in the decision-first group than in the control group.

Table 2: Experiment 1 First and Last Decision Aid Recommendations Compared

	Control	Decision-first
n	14	17
First aid recommendation M	63	118
Median	48	31
Std. Dev.	91	193
Last aid recommendation M	38	39
Median	35	24
Ration of first to last use		
Median	1.00	1.22
Mean	1.86	4.09 *
Std. Dev.	3.05	6.88

* $p < .05$ Mann-Whitney Test

The difference in the proportion of participants that obtained a decreased decision aid recommendation as they iterated through the decision aid is significantly greater in the decision-first group than in the control group, as expected (Table 3). Any participant who accepted the initial decision aid recommendation (total iterations =1) was excluded to avoid including participants that did not pursue the task seriously. The proportion that accepted the initial recommendation did not

significantly differ between groups. In all but one case, the decision-first participants who increased or decreased the sample size moved the decision aid recommendation toward the sample size decision they indicated before using the aid. Because of the small expected cell-size of the sample in Table 3, the proportion was also tested using Fishers Exact Test. The three x two matrix in Table 3 was transformed into a two x two matrix by combining the Increase and Even rows. The results were marginally significant ($p < .07$).

Table 3: Experiment 1 Count of Participants' Direction of Change of Decision Aid Recommendation from First to Last Decision Aid Use

Change Direction	Control	Decision First
Increase	3	1
Even	4	2
Decrease	4	10
Chi-squared = 9.282, p <.01		

The second part Hypothesis 1 regarded the users' evaluation of the decision aid. Participants answered a question with a seven point response indicating agreement with the decision aid, with higher values indicating greater agreement. As expected, the decision-first group (4.56) agreed less with the decision aid than the control group (5.47)(Mann-Whitney U 86.5, $p < .05$).

Hypothesis 2 states the expectation that experienced and confident auditors would direct the decision aid to a lower sample size and agree with the aid less than in-experienced and unconfident auditors. Experience was measured by the number of months audit experience indicated by participants. Confidence was measured by asking how capable the auditors believed themselves to be on a one-to-seven scale. Experience and confidence were significantly correlated with each other, but neither was correlated with the ratio change between first and last decision aid recommendations and agreement with the decision aid; therefore Hypothesis 2 was not supported.

Despite a lack of linear correlation between experience and ratio change, experienced participants demonstrated different behavior than inexperience participants. The auditors whose decision aid recommendations remained even had an average of 49 months experience and stepped through the decision aid an average of 6.33 times. Subjects who increased or decreased the decision aid recommendation had an average experience of 23 and 27 months, respectively, and stepped through the decision aid an average of 3.88 and 4.25 times. More experienced auditors were more likely to experiment with the aid, but were probably more efficient in calibrating the aids inputs with its output.

We also found that participants who directed the recommended sample size either up or down were more uneasy about their decision than those who maintained the decision aid recommendation constant. Those participants who maintained the decision aid's recommendation

experienced less unease (6.3 on a 7 point scale, with uneasiness increasing with low scores) than those who either increased (5.2) or decreased (5.0) the decision aid recommendation from first to last iteration.

EXPERIMENT 2

The second experiment tested whether decision aid guidance decreases the extent that people direct decision aid outcomes to confirm beliefs. The experiment consisted of a between-subjects structure including three groups: a) control, b) rule explanation, and c) justification explanation. Students in a MBA course were offered extra credit to complete the task and were randomly assigned to the three groups. Sixty-six students began the experiment; 21 were eliminated because they did not complete the web-based task. The average age of participants was 28 and the average work experience was 6 years. Age and experience were not significantly different across experimental conditions. Exactly one-third of participants were female; an equal ratio held across all experimental conditions.

The task consisted of two separate stages. In the first stage, the participants read a case about a business purchasing an information system. They were given fictional information about four systems and instructed to pick one. The second stage was administered between one and five days after the first stage, and consisted of reading a second case. The participants were informed that the business in the second case had heard about the decision made in the first case and requested their help in selecting an information system. The first case was used to commit them to a decision; the second case tested subjects' belief confirmation.

After reading the second case, the subjects used a decision aid that recommended an information system based on scores supposedly assigned to the information systems by an independent consultant. The web page containing the decision aid reminded them of their original decision. The decision aid made an initial recommendation based on three out of the possible ten information attributes. The participants were free to change which three attributes the system considered and to calculate a new recommendation as often as they wanted. The final decision aid recommendation was the users' recommendation to the second company. In contrast to the first experiment, participants were able to direct the aid's recommendation by selecting which attributes were used by the aid rather than by changing the values of those attributes. When they were finished, they clicked on a "Finish" button and were asked to confirm that the last decision aid recommendation was their choice for the second case. They then advanced to a debriefing questionnaire.

The control group used a decision aid that initially selected three attributes and recommended an information system. The values assigned to the attributes were concealed. The users could change the attributes, as long as three remained selected. The decision aid in the rule explanation group was the same as that used by the control group with two additions. First, the formula used by the decision aid was revealed to the users (the formula was a simple addition of the

assigned score of each of the three selected attributes). Second, the decision aid displayed the attribute scores. The justification explanation group used the same decision aid used by the control group except that it provided an explanation of why the decision aid selected the initial three attributes used to calculate a recommendation. For example, a message at the top of the decision aid said "The chosen attributes are associated with the cost effectiveness of a system. Systems that have a long-term positive return on investment tend to be more successful. Cost effectiveness includes both system cost and system efficiency. In other words, cost effectiveness leads to system success."

The decision aid recommendation of the participants' final iteration using the decision aid was compared to both a) their recommendation made after reading the first case (hereafter called "prior"), and 2) the decision aid's initial recommendation for the second case (hereafter called "DA"). Because the initial decision aid recommendation was randomly assigned, in eight cases it was the same as the participants' first case recommendation. These cases were removed from the analysis.

When a decision aid explains its recommendation, the aid's persuasiveness is expected to be higher than a decision aid without explanations and users should be more willing to accept the decision aid outcome without directing the decision aid recommendation toward their prior decision. Indeed, the proportion of the explanation groups whose final decision aid recommendation agreed with the decision aid recommendation was significantly higher than for the control group (Table 4, Panel C). The decision aid explanation of its recommendation appears to have convinced some users to adopt the decision aid recommendation. The proportion whose final recommendation was consistent with their initial decision did not differ from the control group, however (Table 4, Panel B). Thus, Hypotheses 3A was supported, but Hypothesis 3B was not. A higher proportion of participants in the control group obtained a response different than both the participant's prior decision and the initial decision aid recommendation (Table 4, Panel A).

SUMMARY

Most decision aid reliance research has used aids that rely upon inputs from the experimenter and require experiment participants to either accept and reject the decision aid recommendation. The experiments conducted in this study, in contrast, expanded upon research by Kachelmeier and Messier (1990) using an aid using subjective inputs and allowing users to direct the decision aid recommendation and found evidence that user direction of decision aids can be explained using the theory of cognitive dissonance, and that decision aid guidance partially offsets direction of decision aids. Both experiments found that many decision aid users, when given the opportunity, will direct the decision aid. Participants in both experiments directed the decision aid outcome toward their prior belief. In experiment 1, two out of the three measures tested found that participants simply stating a conclusion before using the decision aid intensified the tendency to direct the decision aid. That the third measure failed is not surprising given the small sample size and the large variance within the sample.

Table 4:: Experiment 2 Final Decision Pattern			
Panel A: Count of participants directing aid			
Group	Toward Decision Aid	Toward Prior Choice	Away from Both
Control	1	4	5
Rule Listing	6	3	5
Justification	5	5	3
Control vs. experimental groups: chi-squared = 28.48 p<.01			
Panel B: Proportion of participants agreeing with prior choice			
Group	Agree	Disagree	Proportion Agree
Control	4	6	.400
Rule Listing	3	11	.214
Justification	5	8	.385
Control vs. experimental groups: chi-squared = 1.21, n.s.			
Panel C: Count of participants agreeing with decision aid recommendation			
Group	Agree	Disagree	Proportion
Control	1	9	.100
Rule Listing	6	8	.357
Justification	5	8	.385
Control vs. experimental groups: chi-squared = 29.93, p< .01			

Given prior research and the assumption that higher confidence is associated with greater commitment to a decision, the finding that more experienced and more confident participants did not direct decision aids more than their less-experienced and less-confident peers is somewhat surprising. As less-experienced auditors are more likely to use a decision aid, it is significant to note that they tend to direct the decision aid used in this study in the same manner as more experienced auditors. It is possible that the auditors participating in the study form the commitment to a reasonable sample size early in their careers. As would be expected, participants who directed the decision aid toward their stated opinion tended to trust the decision aid less and to be less satisfied with their own decision process.

The experimental instructions may have induced the decision-first group to apply more effort in the decision and to thereby obtain different results than the control group. For example, making a decision with consequences increases effort expended relative to making a judgment without

consequences (Bukszar, 2003). Both groups applied the same effort to using the decision aid, as shown by the number of iterations they processed the aid, however. If their effort was offline from the aid in attending more to the aid outputs, than the question becomes what motivated the decision-first group to expend more effort. The motivation would have to arise asserting a sample size before using the aid, which suggests in increase in decision commitment as the underlying factor.

The results provide evidence that providing decision guidance, whether a simple description of the method used or an explanation arguing in favor of the decision aid recommendation increases the extent that users accept the decision aid's recommendation. The findings parallel prior research in finding that decision aid guidance increases acceptance of a decision aid recommendation; this increased acceptance, however, did not decrease the extent that users direct the decision aid recommendation to their prior decision. Rather, decision aid guidance reduced the proportion of aid users who selected an outcome different from either the decision aid or their own prior decision.

The results also show that decision aid users use decision aids to explore alternative decisions, whether they accept the decision aid recommendation or not. This finding is similar to other research that has shown that decision makers prefer to evaluate what-if scenarios even if they do not improve performance (Kottemann, Davis, and Remus, 1994). Enabling decision makers to explore alternative consequences to their judgments may serve a valuable purpose in decision aid use; if so, decision aid implementers would not wish to limit how decision aid users direct aid recommendations. Often decision aid implementers may wish to restrict aid recommendations, however, or direct decision aid users decisions or processes though use of an aid (Silver, 1990). When decision aids allow user direction of recommendations, the aid is unlikely to change decisions from what a user would obtain unaided, a finding that increases with commitment to the unaided decision. This study's findings suggest two strategies for encouraging reliance on the decision aid outcome. First, the decision process may decrease commitment to the decision. For example, the decision process may emphasize the parameter judgments used by the aid rather than the final decision decision. This emphasis on the parameters may remove some motivation to direct the outcome of the global decision. Organizations may even consider separating the parameter judgments from the decision to limit bias of the parameters. Second, organizations may implement guidance within the decision aid itself to increase the balance of evidence supporting unbiased use of the aid and thereby overcome user commitment an unaided decision.

The findings must be tempered by several observations. The experiment was conducted on the Internet, an environment which may have influenced the results because of a lack of control and may have led to a bias in the results based on those who choose to participate. The results must be interpreted conservatively when applied outside of the specific decision contexts conducted within this study. The two experiments themselves were not based on the same task and their findings may not apply to each other. Further research may increase the ability to generalize the findings.

Participants in this study used the decision aid recommendation as their own decision. Further studies may find whether people direct decision aid outcomes even when they are able to override the decision aid. If so, that finding would indicate that the decision aid recommendation

is directed not only to obtain a result that is desirable to the user, but also to confirm the users' prior beliefs to themselves or to justify the decision to others. In addition, future research may identify whether a change of focus to the input parameters of a decision aid or some form of training can reduce the extent of decision aid direction. Frameworks may be developed to indicate under which circumstances user direction of decision aids is desirable. In conclusion, decision aid direction appears to be associated with users' commitment to their prior decision. Identification of commitment with decision aid directions may provide guidance to remedies, such of decision guidance, which may increase the impact of decision aid recommendations to users or lower users' commitment.

REFERENCES

Arkes, H. R., Dawes, R. M. & Christensen, C. (1986). Factors influencing the use of a decision rule in a probabilistic task. *Organizational Behavior and Human Decision Processes,* 37: 93-110.

Aronson, E. (1968). Dissonance theory: Progress and problems. In R.P.Abelson, E. Aronson, W. J. McGuire, T. J. Newcomb, M. J. Rosenberg & P. H. Tannenbaum (Eds.), *Theories of cognitive consistency: A sourcebook* (pp. 5-27). Chicago: Rand McNally.

Ashton, R. H. (1974). An experimental study of internal control judgments. *Journal of Accounting Research,* 12: 143-157.

Ashton, R. H. (1990). Pressure and performance in accounting decision settings: Paradoxical effects of incentives, feedback and justification. *Journal of Accounting Research,* 28: 148-180.

Audit sampling. (1983) New York: American Institute of Certified Public Accountings.

Audit sampling. (1999) New York: American Institute of Certified Public Accountings.

Brehm, J. W. & Cohen, A. R. (1962). *Explorations in cognitive dissonance*. New York: John Wiley and Sons.

Brock, T. C. & Balloun, J. C. (1967). Behavioral receptivity to dissonant information. *Journal of Personality and Social Psychology,* 6: 413-428.

Bukszar, E. (2003). Does overconfidence lead to poor decisions? A comparison of decision making an djudgment under uncertainty. *Journal of Business and Management*, 9: 115-136.

Cook, G. J. & Swain, M. R. (1993). A computerized approach to decision process tracing for decision support system design. *Decision Sciences,* 24: 931-952.

Davis, E. B. (1998). Decision aids for going-concern evaluation: Expectations of partial reliance. *Advances in Accounting Behavioral Research,* 1: 33-59.

Dawes, R. M., Faust, D., & Meehl, P. E. (1989). Clinical versus actuarial judgment. *Science, 243,* 1668-1674.

Eining, M. M., Jones, D. R. & Loebbecke, J. K. (1997). Reliance on decision aids: An examination of auditors' assessment of management fraud. *Auditing: A Journal of Practice and Theory,* 16: 1-19.

Elliot, A. J. & Devine, P. G. (1994). On the motivational nature of cognitive dissonance: Dissonance as psychological discomfort. *Journal of Personality and Social Psychology,* 67: 382-394.

Festinger, L. (1957). *A theory of cognitive dissonance*. Stanford, CA: Stanford University Press.

Frey, D. (1986). Recent research on selective exposure to information. *Advances in Experimental Social Psychology,* 19: 41-80.

Gibbins, M. (1984). Propositions about the psychology of professional judgment in public accounting. *Journal of Accounting Research,* 22: 103-125.

Gibbins, M. & Emby, C. (1984). Evidence on the nature of professional judgment in public accounting. *Auditing Research Symposium 198*: 181-212.

Gill, T. G. (1995). Early expert systems: Where are they now? *MIS Quarterly,* 19: 51-81.

Glover, S. M., Prawitt, D. F. & Spilker, B. C. (1997). The influence of decision aids on user behavior: Implications for knowledge acquisition and inappropriate reliance. *Organizational Behavior and Human Decision Processes,* 72: 232-255.

Jecker, J. D. (1963). Conflict and dissonance: A time of decision. In R.P.Abelson, E. Aronson, W. J. McGuire, T. J. Newcomb, M. J. Rosenberg, and P. H. Tannenbaum (Eds.), *Theories of cognitive consistency: A sourcebook* (pp. 571-576). Chicago: Rand McNally.

Kachelmeier, S. J. & Messier, W. F. (1990). An investigation of the influence of a nonstatistical decision aid on auditor sample size decisions. *The Accounting Review,* 65: 209-226.

Kaplan, S. E., Reneau, J. H. & Whitecotton, S. M. (2001). The effects of predictive ability information, locus of control, and decision maker involvement on decision aid reliance. *Journal of Behavioral Decision Making,* 14: 35-50.

Kiesler, C. & Pallak, M. (1976). Arousal properties of dissonance manipulations. *Psychological Bulletin,* 83: 1014-1023.

Kiesler, C. & Sakumura, J. (1966). A test of a model for commitment. *Journal of Personality and Social Psychology,* 3: 349-365.

Klienmuntz, B. (1990). Why we still use our heads instead of formulas: Toward an Integrative Approach. *Psychological Bulletin*, 107,3:296-310.

Kottemann, J., Davis, F. & Remus, W. (1994). Computer-assisted decision making: Performance, beliefs, and the illusion of control. *Organizational Behavior and Human Decision Processes*, 57: 26-37.

Lord, C. G., Ross, L. & Lepper, M. R. (1979). Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence. *Journal of Personality and Social Psychology,* 37: 2098-2109.

Messier, W. F., Kachelmeier, S. J. & Jensen, K. (2001). An experimental assessment of recent professional developments in nonstatistical audit sampling guidance. *Auditing: A Journal of Practice and Theory,* 20: 81-96.

Prange, G. W., Goldstein, D. M. & Dillon, K. V. (1982). *Miracle at Midway*. New York: Penguin Books.

Shelton, S., Whittington, O. & Landsittel, D. (2001). Auditing firms' fraud risk assessment practices. *Accounting Horizons*, 15: 19-23.

Silver, M. S. (1990). Decision support systems: Directed and nondirected change. *Information Systems Research*, 1: 47-70.

Staw, B. M. (1976). Knee-deep in the big muddy: A study of escalating commitment to a chosen course of action. *Organizational Behavior and Human Decision Processes,* 16: 27-44.

Sweeney, P. D. & Gruber, K. L. (1984). Selective exposure: Voter information preferences and the Watergate Affair. *Journal of Personality and Social Psychology,* 46: 1208-1221.

Todd, P. A. & Benbasat, I. (1994). The influence of DSS on choice strategies: An experimental analysis of the role of cognitive effort. *Organizational Behavior and Human Decision Processes,* 60: 36-74.

Whitecotton, S. M. (1996). The effects of experience and confidence on decision aid reliance: A Causal Model. *Behavioral Research in Accounting,* 8: 194-216.

Wicklund, R. A. & Brehm, J. W. (1976). *Perspectives on cognitive dissonance*. New York: John Wiley and Sons.

Ye, L. R. & Johnson, P. E. (1995). The impact of explanation facilities of user acceptance of expert systems advice. *MIS Quarterly,* 19: 157-172.

ACQUIRING DOMAIN KNOWLEDGE OF INFORMATION SYSTEMS: THE INFORMATION SYSTEM UPON INFORMATION SYSTEMS APPROACH

Thang Le Dinh, University of Moncton
Gérard Fillion, University of Moncton

ABSTRACT

Nowadays, knowledge is viewed as a significant asset for organizations. Consequently, knowledge management has become an important factor to take into account within and between organizations. This paper proposes an approach for acquiring domain knowledge of information systems (IS) based on conceptual specifications. It is argued that the development process of information systems may include several development and maintenance projects, which could be carried out in parallel, and each project may use its own software process and development method. In order to support the different activities of the IS development process, it is suggested that the development process itself needs a knowledge architecture for managing its domain knowledge. To that end, we propose a knowledge architecture called Information System upon Information Systems (ISIS). An ISIS is considered as a new knowledge-based infrastructure which coexists with other IS infrastructures. It aims at managing the IS domain knowledge. Knowledge management involves activities such as acquiring, analyzing, preserving, and using knowledge. In the present paper, we suggest an approach for acquiring IS domain knowledge, including its identification and organization.

INTRODUCTION

Nowadays, knowledge is viewed as a significant asset for organizations. Consequently, knowledge management has become an important factor to take into account within and between organizations. Activities of knowledge management include acquisition, analysis, preservation, and application of knowledge. Knowledge management allows an organization to leverage its information resources and knowledge assets by remembering and applying experience (Jetter et al., 2006; Watson, 2003).

Domain knowledge is defined as the knowledge of the area to which a set of theoretical concepts is applied and is fundamental to all disciplines (Alexander, 1992). Concerning the information systems (IS) discipline, the term "IS domain knowledge" has dual significance (Khatri

et al, 2006): (a) providing representations, methods, techniques, and tools that form the basis for the development of information systems; (b) providing the solutions for real-world problems in a given business area using those information systems.

The development process of information systems is becoming more and more complex. It may include several development and maintenance projects that are at different stages of their life cycles. Projects may be carried out in parallel and in a distributed environment, and they may employ different software processes, methods, and tools. This tendency offers numerous potential advantages, but also leads to certain challenges regarding the capacity to manage and coordinate effectively such environments.

In our research, we address the challenge of managing the development process using IS domain knowledge. Our work aims at proposing a theoretical framework for a knowledge architecture that supports the management of IS domain knowledge. The knowledge architecture is performed by a typical information system. We called that information system: *Information System upon Information Systems* (ISIS) (Le Dinh, 2004).

An ISIS can be viewed as a new knowledge-based infrastructure that coexists with other IS infrastructures. The ISIS helps an organization to handle its IS domain knowledge, including acquiring, analyzing, preserving, and using domain knowledge.

Furthermore, conceptual specifications of the business domains are tied to play an important role in the construction of deliverables, which are consumed or produced in the IS development process (Moody, 1998; Witt & Simsion, 2004). Conceptual specifications can be found in requirement and design documents, in implemented systems, as well as in user manuals. In fact, we consider that conceptual specifications, which represent the semantic content of information at the conceptual level, are the key information resources used in the IS development process. Consequently, our approach considers conceptual specifications as essential domain knowledge about the IS development process that could be employed to manage this process.

The paper focuses on acquiring IS domain knowledge based on conceptual specifications. By acquiring domain knowledge, we mean to control domain knowledge by identifying and organizing knowledge. The remaining of the paper is structured as follows: first, we present a brief literature review; second, we identify different categories of domain knowledge; third, we discuss the organization of this knowledge; and we end the paper with some conclusions and future directions.

LITERATURE REVIEW

Conceptual Modeling

Although a number of studies argue that domain knowledge has some important impacts on IS problem-solving effectiveness (Curtis et al., 1988; Glass & Vessey, 1992; Khatri et al., 2006),

few studies investigated the role of an important subset of IS development: *conceptual modeling* in domain knowledge management.

IS researchers have shown that conceptual modeling has a great impact on system development costs, flexibility, and ability to meet users' requirements (Moody, 1998; Witt & Simsion, 2004). Prior studies in the area of conceptual modeling have focused on two aspects (Khatri et al., 2006): (1) *Development of a conceptual schema*: involves developing a schema as a representation of an application domain based on users' requirements. This schema should capture the important feature of the information system (Batini et al., 1992; Kim & March, 1995); and (2) *Understanding of a conceptual schema*: assumes existence of a schema and accesses schema understanding. Thus, conceptual schemas can be used to support deep-level cognitive processing by their users (Bodart et al., 2001).

Our research focuses on the latter. We consider that *conceptual* specifications can help IS professionals to access schema understanding. Each conceptual specification focuses on recognizing and representing an informational concept from business domain so that this concept could be incorporated into the information system. Each informational concept concerns an aspect of human perceptions of the real world that may represent the structure of information, the transformation of information, or the coherence of information (Le Dinh, 2004).

For this reason, we consider that conceptual specifications are essential IS domain knowledge. In order to acquire IS domain knowledge based on conceptual specifications, this paper aims at answering the following questions: What informational concepts should be considered part of IS domain knowledge? How those informational concepts are related? and Why those informational concepts are valid?

Knowledge Management

In the following, we discuss similar approaches concerning the management of domain knowledge. These approaches are classified according to three categories: (1) *knowledge management approaches* which concern the knowledge about development projects; (2) *artefact management approaches* which concern the artefacts produced and consumed by the activities of development projects; and (3) *metadata management approaches* which concern the information about business data stored in information systems.

Knowledge management approaches concern how to identify, represent, validate, and access the knowledge about development projects. The popular approaches are *LaSSIE*, *CODE4*, and *ConceptBase*. The LaSSIE system (Devanbu et al., 1991) provides a knowledge-based explicitly representation of information about a large software project, and an interactive interface allowing IS professionals to access directly to the information. The CODE4 system (Skuke & Lethbridge, 1995) is a general-purpose knowledge management system intended to assist the common knowledge processing needs of anyone who desires to analyse, store, or retrieve conceptual knowledge in software development, in the construction of term banks, or in the development of

ontology for natural language understanding. ConceptBase (Jarke et al., 1995; Jeusfeld et al., 1998) is a typical *Knowledge-Based Management System* that aims at supporting the cooperative development and evolution of information systems. Indeed, ConceptBase is the implementation of the Telos language (Mylopoulos et al., 1990) and based on the integration of deductive and object-oriented technologies.

Metadata management approaches aim at providing a repository to manage, validate, use, and evolve metadata stored in tools and systems. There are two categories: *data dictionary* and *metadata repository*. Data dictionary provides a centralized repository of information about database to assist database administrators and developers in managing, controlling, and validating the collection and the use of data. Metadata repository aims at integrating and managing specifications using standard metadata models and interchange format standards (Marco, 2000). There are two competitive organizations for metadata standards: the *Meta Data Coalition* (MDC) with *Open Information Model* (Bernstein et al., 1999) and *Meta Data Interchange Specification* (MDIS, 1997); and the Object Management Group (OMG) with *Common Warehouse Metamodel* (CWM, 2003), *Meta Object Facilities* (MOF, 2002) and *eXtensible Metadata Interchange* (XMI, 2000).

Artefact management approaches are primarily designed to serve as repositories of all the artefacts produced and consumed during the life cycle of an IS development project. These artefacts can be schema, documentations, programs, reports, and tools that intend to provide support for all activities of IS development, from requirement analysis to system implementation. The typical approaches are the *DAIDA environment*, *Software Information base*, and *Software Configuration Management*. *DAIDA* is an environment, developed in an ESPRIT project, which is based on the State-Of-The-Art languages for requirements modelling, design, and implementation of information systems (Jarke et al., 1992). *Software Information Base* (SIB) is a software repository system that provides organization, storage, management, and access facilities for reusable software artefacts (Constantopoulos et al., 1995). The SIB supports the representation of information about requirements, designs, and implementations of software artefacts. *Software Configuration Management* provides facilities for identifying, monitoring and controlling configuration, managing, as well as supporting teamwork (Stephen & Brad, 2002).

The Information System upon Information Systems Approach

The differences between the *Information System upon Information Systems* (ISIS) approach and the other approaches can be derived from the main characteristics of the ISIS approach: considering conceptual specifications as essential IS domain knowledge; representing specifications in a coherent manner, including the structure, behaviour and coherence of specifications; providing the capacity to work with a subset of specifications as parts of an IS; and supporting the management and coordination of specifications throughout all phases of the IS development process.

Therefore, the major differences of the ISIS approach with the other approaches are the following. Firstly, there is a little focus on representing the behaviour of knowledge. In the ISIS

approach, the behaviour of knowledge is represented by the possible dynamic states of specifications and the transitions between those dynamic states. Secondly, in the ISIS approach, the coherence of knowledge is represented by a set of quality rules, for example the completeness and validity rules. However, the related approaches are not concerning very much the quality rules. Thirdly, the ISIS approach aims at supporting the management of domain knowledge throughout the IS development process, meanwhile some approaches focus only on some phases of the development process. For instance, the Metadata repository systems aim at the design phase; the Data dictionary approach aims at the implementation and exploitation phases; the Knowledge management approach aims at supporting the conceptual specifications at the requirement and design phases. Fourthly, there is not so much concern about supporting the coordination of domain knowledge within a development team and between different teams working in parallel. Moreover, the coordination of specifications in the industrial approaches is not so flexible and generally based on the *check-in check-out* mechanism. In the ISIS approach, the coordination is based on the information overlaps and overlap protocols to manage these overlaps. Lastly, there is a little focus on working with parts of an IS. There is only the DAIDA approach that deals with the structured and complex objects to represent artefacts. This approach also provides the facility to represent objects in a modular way.

To summarize, the specificity of the ISIS approach is to provide a knowledge-based architecture that supports management and coordination of IS domain knowledge based on conceptual specifications *in depth* and *in width*. This architecture aims at constructing the *Information layer* as a new IS infrastructure that manages the structure, behaviour, and coherence of domain knowledge of information systems.

IDENTIFICATION OF DOMAIN KNOWLEDGE

As mentioned above, we consider that conceptual specifications are essential IS domain knowledge. Thus, identification of domain knowledge consists to identify different categories of conceptual specifications.

Several studies have sought to show IS professionals how to identify and model systems using different model concepts (Wand et al., 2002): things, objects, entities, and properties/attributes (e.g., Wand et al., 1993); classes and class structures (e.g., Parsons, 1996); part-of relationships (e.g., Storey, 1991); events, processes, and workflows (e.g., Basu & Blanning 2000; Curtis et al., 1992); decompositions and level structures (e.g., Paulson et al., 1992); and business rule and integrity constraints (e.g., Ross, 2005).

However, according to us, there is a little attention to examine all the categories of conceptual specifications in a global and coherent manner. A global and coherent manner means that all aspects of human perceptions of the real world must be taken into account in order to answer the questions *What*, *How*, and *Why* as mentioned in the previous section.

Indeed, the IS development is actually related to modeling, specifying, as well as realizing the informational concepts from the business domain. Informational concepts may represent:

structure of information, such as the "Invoice" informational concept; or *transformation of information*, such as the "Paying an invoice" informational concept; or *coherence of information*, such as a business rule concept that states: "the total amount of all the payments of an invoice must not exceed the amount of that invoice".

Therefore, we argue that there are three aspects of information systems: (1) the *static aspect* concerning the structure of information; (2) the *dynamic aspect* concerning the transition of information; and (3) the *integrity rules aspect* concerning the coherence of information.

Besides, an information concept from the business domain may be represented by an IS concept from the IS domain. An IS concept must belong to the *static aspect*, the *dynamic aspect*, or the *integrity rule aspect* of an IS.

Despite the diversity in IS development methods and methodologies, there are some IS concepts that are invariant (for example, the "class"). Those concepts are called *key concepts*. Each key concept corresponds to a certain type of conceptual specifications.

In the following, we present the *key concepts* corresponding to the static, dynamic, and integrity rules aspects. The major part of these concepts could be found in object-oriented methods, such as Object-Oriented Analysis (OOA) (Coad & Yourdon, 1990), Object-Oriented Design (OOD) (Coad & Yourdon, 1991), Grady Booch (Booch, 1991), Object Modeling Technique (OMT) (Rumbaugh et al., 1991), Shlaer and Mellor (Shlaer & Mellor, 1989, 1992), Object-Oriented Software Engineering (OOSE) (Jacobson, 1992), and Unified Modeling Language (UML) (Booch, 1998; Rumbaugh et al., 1998).

In summary, this section aims at presenting the different categories of IS domain knowledge in a global and coherent manner. First, IS domain knowledge will be classified and studied according to the three aspects of information systems: static, dynamic, and integrity rules. Second, key concepts of each aspect will be presented; each concept is corresponding to a category of IS domain knowledge. Third, the relation between concepts within and between aspects of IS will be also discussed.

Concepts of the Static Aspect

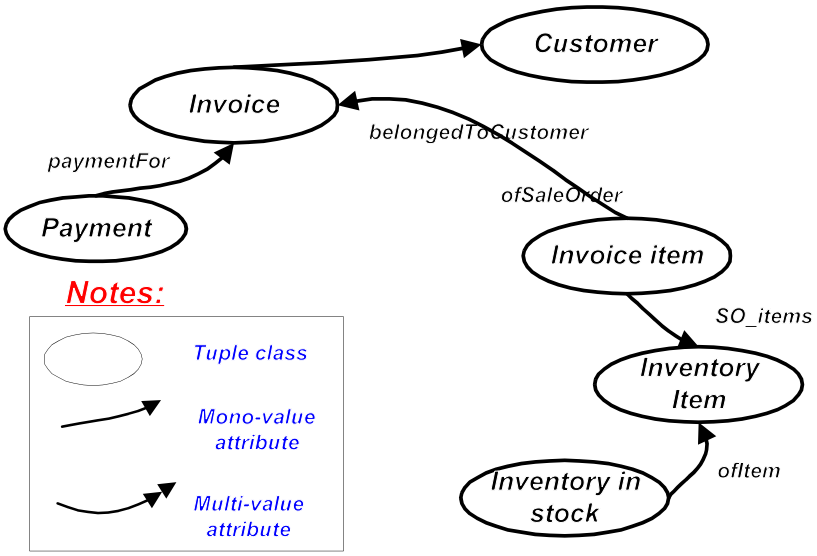
The concepts of the static aspect describe what types of information exist, their structures, as well as their interrelations. They include *atomic-class*, *tuple-class*, *hyperclass*, *attributes*, *key*, and *subclass*.

An object type and a set of objects of this type define a class. In our approach, there are three kinds of classes: An *atomic-class*, which is defined as a primitive class. It is indecomposable. Objects of an atomic-class have a particular characteristic: their identifier is also their value. A *tuple-class*, which contains objects having the same structure and the same behaviour. A structure of a tuple-class is characterized by a set of attributes. The behaviour of a tuple-class is represented by a set of methods. And a *hyperclass*, which is a subset of classes that are all connected without ambiguity by navigation links to a *root class* (Andany et al., 1991; Turki, 2002).

The interrelations between the classes' concepts lead to other concepts such as *attribute, key,* and *subclass*. An *attribute* of a class is a function corresponding to every object of this class and to a set of objects of other classes. A *key* of a class is defined by a set of special attributes which can be used to distinguish one object from other objects in the same class. A class can define its *subclasses*. The interpretation of a subclass is the set of all identifiers of the interpretation of its superclass for which the specialization condition is evaluated to "true".

Example: Let us discuss a typical IS named MATIS used in small enterprises. This example focuses on the Sale/Collection process, particularly the direct sale activities of the enterprise. The information related to this process can be represented by the following tuple-classes: *Customer, Inventory item, Inventory in stock, Invoice, Invoice items,* and *Payment*. Figure 1 introduces these tuple-classes and their attributes.

Figure 1. Tuple-Classes Related to Sale/Collection Process



Concepts of the Dynamic Aspect

The concepts of the dynamic aspect are presented in the two levels of behaviour: *local behaviour*, defined as the behaviour of objects of a class, including the concepts of *dynamic states* and *method*; and *global behaviour*, defined as the behaviour of the IS, including the concepts of *event* and *process*.

The local behaviour is represented by the concepts of *dynamic state* and *method. Dynamic states* of an object are modes or situations during which certain methods are "enabled" and other

methods are "disabled". A *method* of a class is used to transit between the dynamic states of the objects of this class. In other words, a method transfers a set of dynamic states to another set of dynamic states of a class.

The global behaviour is represented by the concepts of *event* and *process*. *Event* is a remarkable phenomenon outside of the IS that may provoke a change of its dynamic states. In fact, the event structure helps to define the interface of the IS with its environments. A *process* is a feedback of the IS to the occurrence of an event. In fact, a process performs a transformation of a set of dynamic states of the IS.

Example: The global behaviour of a sale has been introduced in Figure 2 by a *life-cycle diagram* based on *Petri-Net* (see Peterson, 1981). As shown in Figure 2, three events may occur: when a customer *demands for items* (e_1); when a customer *gives personal information* (e_2); and when a customer *pays for an invoice and receives items* (e_3). These events may trigger their corresponding processes such as *Getting-demand, Checking-stock*, *Getting-customer-info*, *Creating-new-customer* and *Receiving-payment-and-Delivering*.

Concepts of the Integrity Rules Aspect

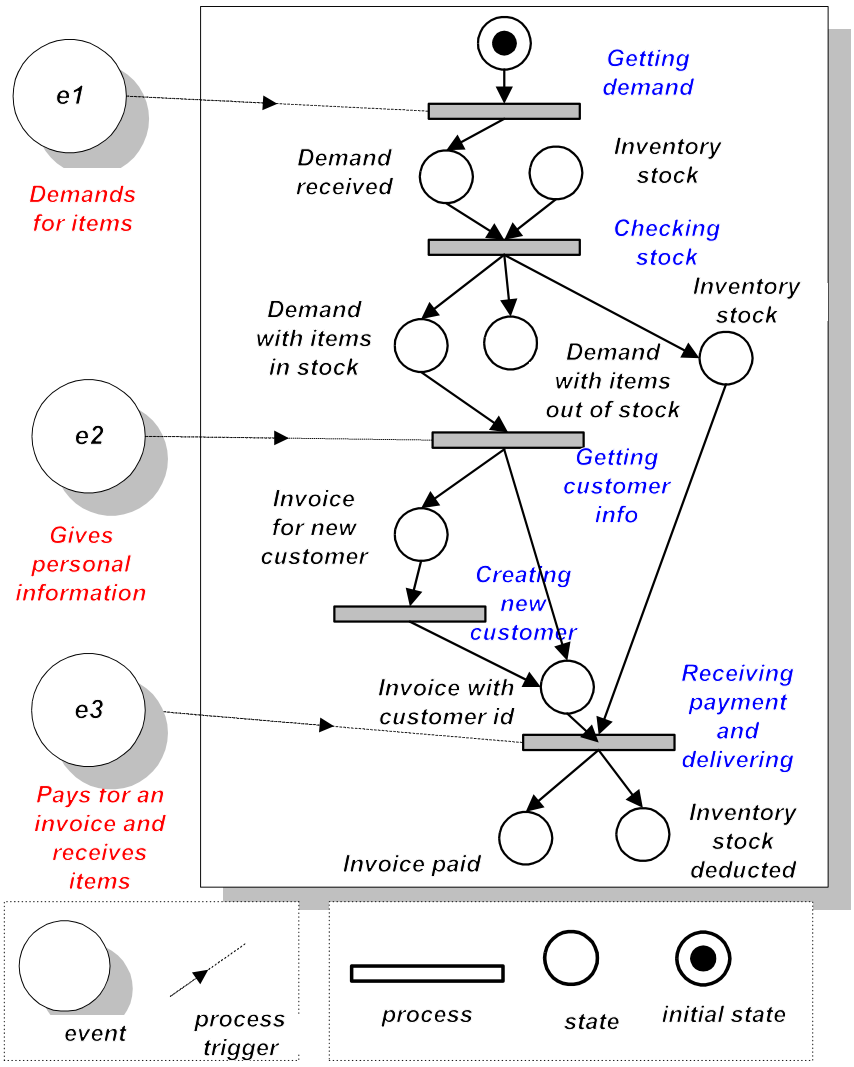
The integrity rule aspect includes the concepts of its main components such as *integrity rule* and *primitive*, as well as about their interrelations such as *scope* and *risk* (Léonard, 2003; Luu & Léonard, 2002).

Integrity rules (IR) represent the business rules of an organization. An IR is actually a logical condition defined over tuple-classes that can be formally specified and verified by processes or methods.

A *primitive* is a basic operation on a tuple-class such as *create*, *update*, and *delete*. The execution of a primitive may violate the validation of an IR. *Scopes* of an IR represent the context of an IR, including a set of tuple-classes on which the IR has been defined. *Risks* are the possibilities of suffering the incoherence of information. In general, a risk is defined on a scope and a primitive. In particular, especially in the case of the *update* primitive, it is indispensable to specify the related attribute of a risk.

Example: This example is related to an integrity rule IR_1 described as: "the quantity of each item of an invoice must not exceed the quantity in stock of this item". The *scopes of* IR_1 are the *Invoice Item* and *Inventory in stock* classes. The IR can be verified by *Checking-stock* process. Its *risks* are the methods for insertion and modification of *Invoice Item*, as well as deletion and modification of *Inventory in stock* classes.

In the next section of the paper, we discuss the organization of this knowledge identified by the means described in the previous sections.



ORGANIZATION OF DOMAIN KNOWLEDGE

Once domain knowledge is acquired, the next step is to apply experience or to apply domain knowledge based on the context of application. This section is then related to explain how to work with a semantic context, particularly a unique and coherent set of IS concepts.

In other words, our approach must allow to work with a part of an IS, in particular a set of conceptual specifications as a semantic unit. We propose the concept of *IS component* that can be used to represent a semantic unit. Each IS component has a particular purpose. Since even the smallest enterprise has multiple purposes with different areas of concerns, it will potentially have many different IS components.

Indeed, it is obvious that the component-based development approaches have been considered as a common solution for IS implementation, deployment, and management. Consequently, many research and industrial works have tried to evolve the component paradigm for enterprise-wide IS development and management (Arsanjani, 2002; Herzum & Sims, 2000; Stojanovic et al., 2001).

For the same reason, the concept of IS component allows to work with a part of an IS (Le Dinh & Léonard, 2002; Léonard et al., 2001) as a component. An IS component can be considered an autonomous unit that will be developed from analysis to deployment levels. In the following, we present the definition of an IS component and then point out how to specify this IS component.

Definition of an IS Component

One of the main challenges for the IS community today is how to manage the development of different IS in parallel, while at the same time being able to adapt rapidly to the evolutions of these systems caused by technology, business, and organizational environment changes. Historically, there was a well-known "Divide and Conquer" solution based on *modularization* and *separation of concerns*, which authorized working in parallel, decentralizing, and making decision (Arsanjani, 2002; Parnas, 1972).

In our view, components for enterprise-wide IS require the addition of another dimension related to the separation of concerns. This new dimension could allow data and its metadata, such as structural and behavioural specifications, to be used to modify the contextual behaviour of components and their implementations at design-time, build-time, and run-time.

An *IS component* is a reusable artefact of an IS part that represents adequate characteristics of an autonomous system (Le Dinh & Léonard, 2002). Indeed, an IS component is a representation of the structure, behaviour, and coherence of information (including specifications and data) within an organization or a part of an organization.

From the external view, an IS component can be considered as an autonomous IS that can be developed and delivered independently from the others. However, autonomy does not mean isolation. An IS component can easily cooperate with other IS components to provide more valuable functionalities. The process of developing IS components is often seen as a process of discovery of pre-existing facts, but it is more useful to see it as a process of negotiation: developing a consensus between different stakeholders within the area of interest.

An IS component is cooperative if it shares goals with other elements in its environment such as other IS components, organizational roles, and the enterprise itself in order to contribute towards the fulfillment of common goals (see Figure 3).

From the internal view, IS components are composed of a set of elements corresponding to the three facets of an IS and their transformations at different levels of abstraction (De Michelis et al., 1998): the *organizational facet* is related to the management of the work from a formal organizational perspective; the *group collaboration facet* is related to the organizational

roles/responsibilities working on common business process; and the *systems facet* is related to the computerized systems that support business activities. Figure 4 provides a brief overview of the different elements of an IS component in accordance with the three facets of an IS.

Figure 3. External View of Information System Components

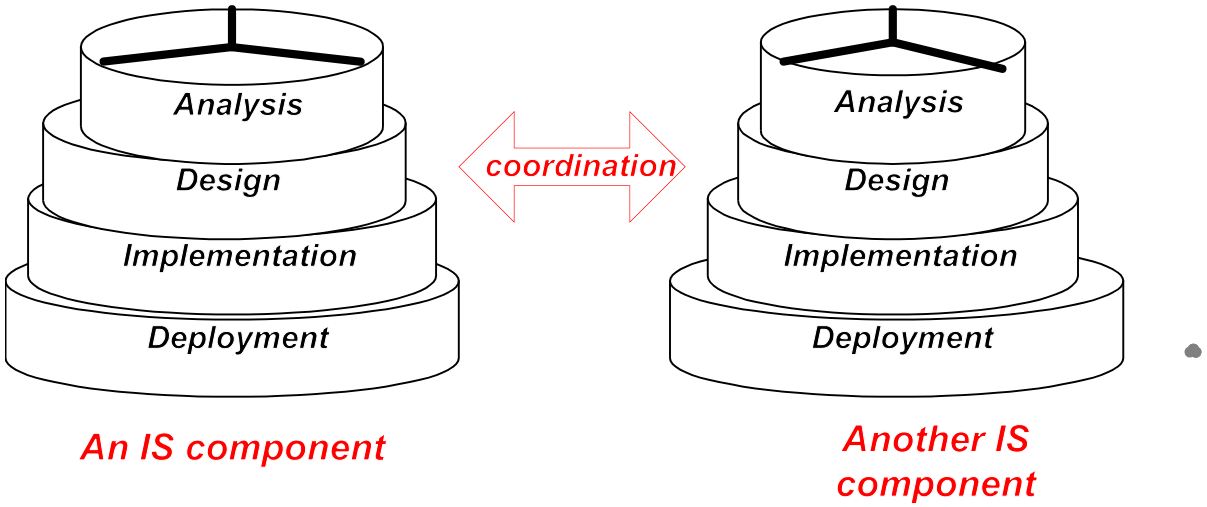
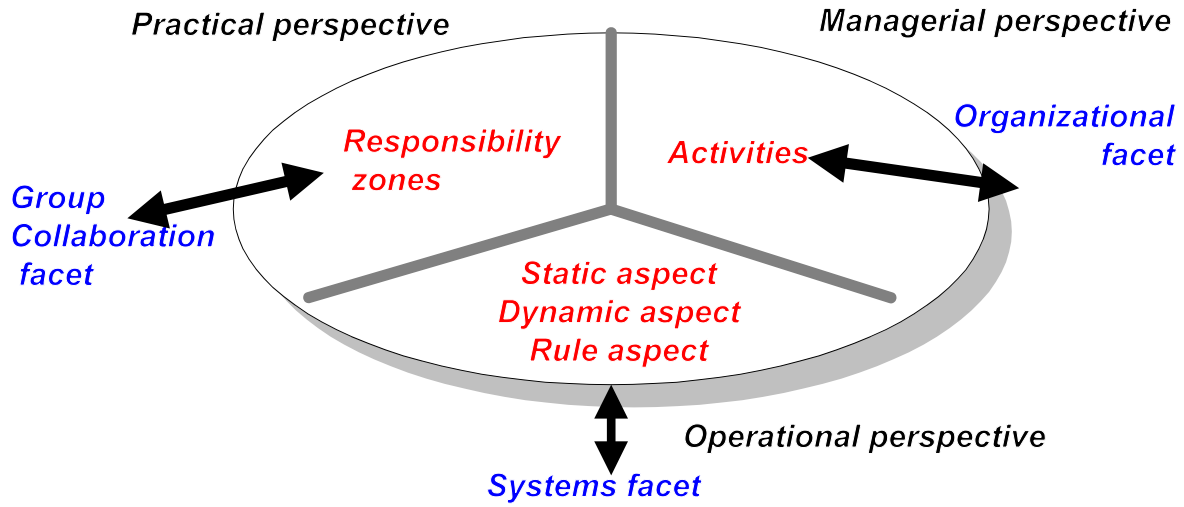


Figure 4. Overview of an IS Component



As shown in Figure 4, there are three standpoints for viewing the work within an organization: the *operational perspective* as executions of operations on a system; the *practical perspective* as events in the history of group practice; or the *managerial perspective* as performances intended to fulfill organizational objectives in accordance with organizational rules.

In our approach, the practical perspective, which is reflected by the group collaboration facet, contains the *responsibility zone* element. The managerial perspective, which is reflected by the organizational facet, contains the *business activities* element. And the operational perspective, which is reflected by the system facet, contains the elements of the *static*, *dynamic*, and *integrity rule* aspects of an IS.

Description of an IS Component

To specify an IS component, we use the *3C model* of reusable software components that was originally proposed by Tracz (see Edwards, 1992) and widely accepted by the computing community. The *3C* stands for the three distinct views of a component: concept, content, and context. *Concept* focuses on the abstraction captured in a component; *content* is the implementation of that abstraction; and *context* deals with the environment in which the component is designed to interact.

Concept of an IS Component

An IS component itself is an autonomous IS and therefore mostly takes part in the System facet. As a result, the concept of an IS component must be constituted by all fundamental aspects of an IS such as the static, the dynamic, and the integrity rules aspects.

Content of an IS Component

As an implementation of the context of an IS component, the content of an IS component must include the detailed descriptions of all the elements of the three fundamental aspects of an IS. In brief, the content of an IS component is represented by: a unique *hyperclass* and its dependents (representing the static aspect); a set of *events* and *processes* defined on that hyperclass and their dependents (representing the dynamic aspect); and a set of *integrity rules*, whose *scopes* defined on that hyperclass, and their dependents (representing the integrity rules aspect).

The formal definition of the content of an IS component is the following:

Definition

Content of an IS component is a tuple

<Hcl, Cl_i, A_j, S_k, M_l, E_m, P_m, Ir_o> where:
Hcl is a Hyperclass;
Cl_1, Cl_2, … Cl_n is a set of n classes that constitutes the hyperclass Hcl;

{A_j: attribute from class Cl_p to Cl_q and p,q Î [1,n]} is a set of attributes;
{S_k: state k of objects of class Cl_i and i Î [1,n]} is a set of dynamic states;
{M_l: method l of class Cl_i and i Î [1,n]} is a set of methods;

E_m is a set of events that may occur to the component ISc and their corresponding processes P_m that can access the set of classes of the hyperclass Hcl; and
Ir_o is a set of integrity rules defined on classes Cl_1, Cl_2, … Cl_n.

Context of an IS Component

The context of an IS component is formed of two types of interactions: (1) the interactions of an IS component with the organizational environment, including the organizational and the group collaboration facets; and (2) the interactions between IS components.

To represent the interactions of an IS component with the organizational environment, we define the concepts of *activities* and *responsibility zones*. *Activities* represent the organizational standpoint of a business process, whereas the corresponding processes of an IS component represent the computerized part of this process. A *responsibility zone* (RZ) is a part of an organization-working environment. In our approach, it is defined by its corresponding IS components and its activities.

Furthermore, information overlap plays the decisive role of the interactions among IS components. Information overlap among components is indispensable when several components share a common subset of information (Léonard, 1999).

In our approach, information overlap can be managed as *overlap situations* and *overlap protocols*.

An *overlap situation* occurs when there is at least one class or one process in common to several IS components. There are possibly three types of overlap situations: *distinct*: there is no common class and no common process between IS components; *with borders*: there are common classes but no common process; and *with overlaps*: there are common classes and common processes.

Depending on the levels of abstraction and the nature of information overlapped, there are two *overlap layers*: (1) *metadata overlap layer* is related to overlaps of the *metadata* of IS

components (i.e. specifications); and (2) *data overlap layer* concerns overlaps of the *data* of IS components (i.e. information). It is important to take into account the semantic heterogeneities that may happen to metadata overlap layer and the data redundancies that may happen to data overlap layer. Various approaches can be used to resolve the semantic mismatch and data redundancy problems (e.g., Batini et al., 1986; Hammer & McLeod, 1993; Hull, 1997; Mirbel, 1997; Parent & Spaccapietra, 1998).

An *overlap protocol* is a protocol that allows each RZ to perform its own processes locally. Meanwhile, this protocol also helps the RZ to take into account the processes in other RZs that can influence its own processes. The protocol must therefore allow the different RZs to process their own activities, while at the same time being aware of proceedings in the other RZs, especially when such proceedings may influence their own activities. In fact, an overlap protocol includes a set of semantics, rules, and formats managing the cooperation between IS components. It is possible to combine several overlap protocols in a coherent way to coordinate the interactions among RZs.

A generic overlap protocol has been proposed by Léonard and Parchet (1999): the *shade projection* overlap protocol. This overlap protocol authorizes a RZ of an IS component to use *shade processes*, which are overlapped with other RZs (of other IS components). With these shade processes, one RZ can take into account the fact that some important activities occur in another RZ and vice versa.

Actually, three techniques can be used to implement overlap protocols: (1) the *ownership-based overlap technique* appoints which one RZ will play the role of the *custodian* for each common element. The custodian of an element takes the responsibility for defining, developing, and maintaining it. The other RZs may communicate to the custodian to obtain information about this element; (2) the *alert-based overlap technique* allows a RZ to send an *alert* to other RZs after performing a process which is overlapped between them; and (3) the *service-based overlap technique* allows a RZ to send a *request* to perform a process to another RZ who is the *provider* of this process. Normally, the *provider* RZ will perform the requested process and return the result to the requested RZ.

The first overlap technique can apply to the "with borders" overlap situations. Meanwhile, the second and the third overlap techniques, which can be used to implement *shade processes*, apply to the "with overlaps" overlap situations.

The formal definition of the context of an IS component is the following:

Definition

Context of an IS component is a tuple <RZ, A_i, $O.S_j$,$O.P_k$ > where:
RZ is a corresponding responsibility zone;
A_1, A_2, ... A_n is a set of corresponding business activities;
$O.S_j$ is a set of overlap situations between this IS component and other IS components; and
$O.P_k$ is a set of overlap protocols to coordinate these overlap situations.

Example Related to IS Components

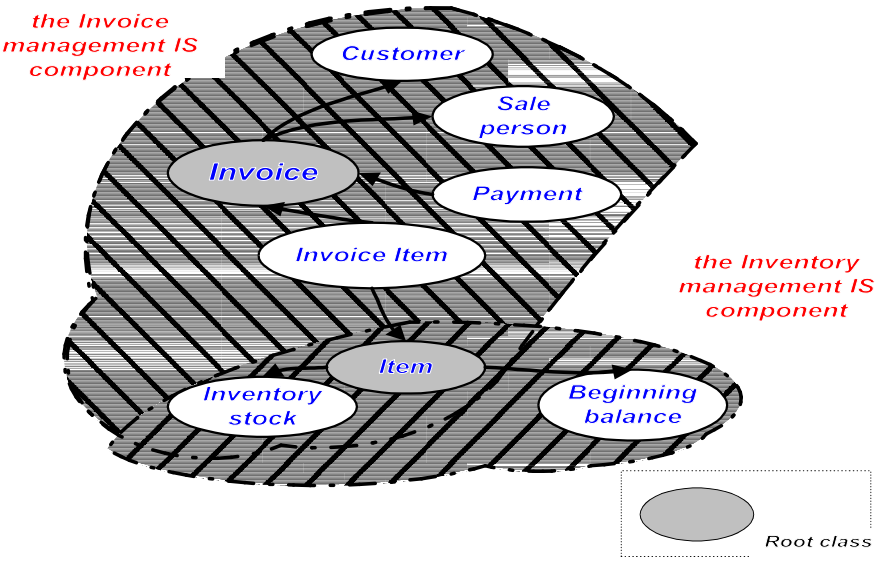
This example illustrates how to describe IS components and information overlaps.

Information System Components

Let us return to the example related to a retail-trading store presented earlier. We define the *Invoice management* IS component aiming at supporting the direct sale activities, which are responsible by the *Sale section*. The static aspect of this IS component is defined on the hyperclass that has the *Invoice* tuple-class as its root class. From the *Invoice* tuple-class, one can navigate to other related tuple-classes such as *Customer, Inventory item, Inventory in stock, Invoice items*, and *Payment* without any ambiguity (see Figure 5). The dynamic aspect of this IS component can be referred to the global life cycle of a sale presented in Figure 2.

Besides, there is another IS component: the *Inventory management* IS component, which has the *Item* tuple-class as its root class. Its other constituents are the *Inventory stock* and *Beginning balance* tuple-classes.

Figure 5. The information Overlap Between the *Invoice Management* and *Inventory Management* IS components



Overlap situations

There are two overlap situations related to these two IS components: os_1 and os_2. os_1 is a "*with border*" overlap situation concerning the *Item* class, which is shared between these two IS components; and os_2 is a "*with overlaps*" overlap situation regarding the *Inventory stock* class, which is shared by the two IS components and both of them may need to perform the same processes on it.

Overlap protocol

In this example, we only focus on the metadata overlap layer. For os_1, we propose to apply the ownership-based overlap technique for the metadata overlap layer. It is assumed that the RZ of the *Inventory management* IS component is the custodian of the *Item* class. As a result, the RZ of this IS component is responsible for creation, modification, and deletion of conceptual specifications of this class. As for the metadata overlap layer of os_2, we propose to apply the alert-based overlap technique. Consequently, both the RZs of *Invoice management* and *Inventory management* IS components can perform the shade processes. However, the RZ of each IS component has to send an alert to the other RZ after performing a shade process (such as creation, modification, or deletion of the specifications) belonged to the overlap zone.

CONCLUSION

In our research, we have shown the importance of a new knowledge architecture based on an *Information System upon Information Systems* (ISIS), which supports the management of domain knowledge used in IS development. Knowledge management involves the acquisition, analysis, preservation, and application of domain knowledge. We also consider that conceptual specifications, which represent the semantic content of information at the conceptual level, are the essential IS domain knowledge.

The perspective of this work is to provide an effective knowledge-based architecture that would be best suited for the management and coordination of IS development and maintenance projects in a complex and distributed environment.

The contribution of this paper is to provide a unique and coherent framework for acquiring knowledge, including the identification and the organization of knowledge based on the IS key concepts. The concepts proposed can be classified according to the different aspects of an IS such as: static aspect: *atomic-class*, *tuple-class*, *hyperclass*, *attributes*, *key*, and *subclass*; dynamic aspect: *dynamic states, method, event*, and *process*; integrity rule aspect: *integrity rule, primitive, scope*, and *risk.* Regarding the group collaboration facet, there are the concepts of *information system component*, *responsibility zone, overlap situation*, and *overlap protocol.*

Concerning the implication of our work for practice, if an organization has several projects carried out in parallel and used different development methods, our work provides guidelines to identify and to organize its domain knowledge based on the proposed categories of specifications.

Concerning the implication for research, much work remains to be carried out to apply the knowledge architecture on a broader scale. We are actually working on analyzing, preserving, and using domain knowledge.

In order to experiment the proposed knowledge architecture, we are also to develop a first prototype in Software-as-a-Service environment. The purpose of this prototype is to support knowledge management based on conceptual specifications. The prototype will allow guarantee some quality factors of conceptual specifications such as completeness, correctness, integrity (Moody et al., 1998). Besides, the prototype will also allow to monitor overlap situations and to realize overlap protocols. In addition, we also have foreseen to extend our prototype to cooperate with external environments, particularly the development tools and systems such as computer-aided software engineering (CASE) and database management system (DBMS).

ACKNOWLEDGMENTS

The authors would sincerely like to thank the *Faculté des Études Supérieures et de la Recherche* (FESR) at the University of Moncton for its financial contribution to this study. We also thank Professor Michel Léonard at the University of Geneva for his comments on the manuscript.

REFERENCES

Alexander, P.A. (1992). Domain knowledge: Evolving themes and emerging concerns. *Educational. Psychology*, *27*(1), 33-51.

Andany, J., M. Léonard & C. Palisser (1991). Management of schema evolution in databases. *Proceedings of I/LDB91*, Morgan-Kaufman Publishers.

Arsanjani, A. (2002). Developing and integrating enterprise components and services. *Communications of the ACM*, October.

Basu, A. & R.W. Blanning (2000). A formal approach to workflow analysis. *Information Systems Research*, *11*(1) 17-36.

Batini, C., M. Lenzerini & S.B. Navathe (1986). A comparitive analysis of methodologies for database schema integration. *ACM Computing Surveys*.

Batini, C., S. Ceri & S.B. Navathe (1992). Conceptual database design: An entity-relationship approach. *Benjamin/Cummings*, ISBN 0-8053-0244-1.

Bernstein, P.A., T. Bergstraesser, J. Carlson, S. Pal, P. Sanders & D. Shutt (1999). Microsoft repository version 2 and the open information model. *Information Systems*, *24*(2), 71-98.

Bodart, E, A. Patel, M. Sim & R. Weber. (2001). Should optional properties be used in conceptual modelling? A theory and three empirical tests. *Information Systems Research*, *12*(4), 384-405.

Booch, G. (1998). *The Unified Modeling Language User Guide*. Addison Wesley.

Booch, G. (1991). *Object Oriented Design with Applications*. Benjamin/Cummings.

Coad, P. & E. Yourdon (1990). *Object-Oriented Analysis*. Prentice-Hall.

Coad, P. & E. Yourdon (1991). *Object-Oriented Design*. Yourdon Press.

Constantopoulos, P., M. Jarke, J. Mylopoulos & Y. Vassiliou (1995). The software information base: A server for reuse. *VLDB Journal*, *4*(1).

Curtis, B., H. Krasner & N. Iscoe (1988). A field study of the software design process for large scale systems. *Communications of the ACM*, *31*(11), 1268-1287.

Curtis, B., M.I. Kellner & J. Over (1992). Process modeling. *Communication of the ACM*, *35*(9), 75-90.

CWM (2003). Common warehouse metamodel (CWM) specification, v1.1. OMG. Retrieved December 2, 2007, from http://www.omg.org.

De Michelis, G., E. Dubois, M. Jarke, F. Matthes, J. Mylopoulos, J.W. Schmidt, C. Woo & E. Yu (1998). A three-faceted view of information systems. *Communications of the ACM*, December.

Devanbu, P.T., R.J. Brachman, P.G. Selfridge & B.W. Ballard (1991). Lassie: a knowledge-based software information system. *International Conference on Software Engineering*.

Edwards, S.H. (1992). Toward a model of reusable software subsystems. *5th Annual Workshop on Software Reuse*.

Glass, R.L. & I. Vessey (1992). Toward a taxonomy of application domains: History. *Journal of Systems Software*, *17*(2), 189-199.

Hammer, J. & D. McLeod (1993). An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *International Journal of Intelligent and Cooperative Information Systems*.

Herzum, P. & O. Sims (2000). *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*. OMG Press.

Hull, R. (1997). Managing semantic heterogeneity in databases: A theoretical perspective. *Symposium on Principles of Database Systems*, 51-61.

Jacobson, I., M. Christerson, P. Jonsson, G. Overgaard (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach (ACM Press*. Addison-Wesley.

Jarke, M., J. Mylopoulos, J.W. Schmidt & Y. Vassiliou (1992). DAIDA: An environment for evolving information systems. *ACM Transactions on Information Systems*, *10*(1), 1-50.

Jarke, M., S. Eherer, R. Gallersdorfer, M.A. Jeusfeld & M. Staudt (1995). Conceptbase - a deductive object base for meta data management. *Journal on Intelligent Information Systems*.

Jetter, A., J. Kraaijenbrink, H.-H. Schröder & F. Wijnhoven (2006). *Knowledge Integration: The Practice of Knowledge Management in Small and Medium Enterprises*. Physica-Verlag.

Jeusfeld, A., M. Jarke, H.W. Nissen & M. Staudt (1998). ConceptBase - Managing conceptual models about information systems. In P. Bernus, K. Mertins, and G. Schmidt. Springer-Verlag.

Khatri .V., I. Vessey, V. Ramesh, P. Clay & S.J. Park (2006). Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. *Information Systems* Research, *17*(1), 81-102.

Kim, Y-G. & S.T. March (1995). Comparing data modeling formalisms. *Communication of the ACM*, *38*(6), 103-115.

Le Dinh, T. & M. Léonard (2002). Defining information system components. *Proceedings of Confederated International Conferences CoopIS, DOA, ODBASE 2002*, Springer, California, October.

Léonard, M., A. Grasset, T. Le Dinh & C. Santos (2001). Information kernel: An evolutionary approach to integrate enterprise information assets. *Conference of Open Enterprise Solutions: Systems, Experiences, and Organizations*, Rome, September.

Léonard, M. (2003). Information system engineering getting out of classical system engineering. Keynote lecture, *5th International Conference On Enterprise Information Systems*, Angers, France, April.

Léonard, M. & O. Parchet (1999). Information overlap. *IEEE International Symposium on Database Applications in Non-Traditional Environments,* DANTE'99.

Luu, H. & M. Léonard (2002). Towards an integrity rules management system. Internal report, Database group, University of Geneva.

Marco, D. (2000). *Building and Managing the Meta Data Repository: A Full Lifecycle Guide*. Wiley.

MDIS (1997). The meta data coalition - Meta data interchange specification (MDIS Version 1.1). Retrieved December 2, 2007, from http://www.tdan.com/view-articles/4138.

Mirbel, I. (1997). Semantic integration of conceptual schemas. *Data Knowledge Engineering* 21.

MOF (2002). Meta object facilities (MOF) specification, Version 1.4. Retrieved December 2, 2007, from http://www.omg.org.

Moody, D.L. (1998). Metrics for evaluating the quality of entity relationship models. Tok Wang Ling, Sudha Ram, Mong-Li Lee (Eds.). *Proceedings of the 17th International Conference on Conceptual Modeling (ER 1998). Lecture Notes Computer Science Springer* (211-225).

Mylopoulos, J., A. Borgida, M. Jarke & M. Koubarakis (1990). Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems*, *8*(4).

Parent, C. & S. Spaccapietra (1998). Issues and approaches of database integration. *Communications of the ACM*, *41*.

Parnas, D.L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12).

Parsons, J. (1996). An information model based on classification theory. *Management Science*, 42(10), 1437-1453.

Paulson, D. & Y. Wand. (1992). An automated approach to information systems decomposition. *IEEE Transactions Software Engineering*, *18*(3), 174-189.

Peterson, J.L. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice Hall.

Ross, R.G. (2005). *Business Rule Concepts ~ Getting to the Point of Knowledge (2nd Edition)*. Business Rule Solutions, LLC.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy & W. Lorensen (1991). *Object-Oriented Modeling and Design*. Prentice-Hall.

Rumbaugh, J., I. Jacobson & G. Booch (1998). *The Unified Modeling Language Reference Manual*. Addison-Wesley.

Shlaer, S. & S.J. Mellor (1989). *Object-Oriented System Analysis: Modeling the World in Data*. Yourdon Press.

Shlaer, S. & S.J. Mellor (1992). *Object Lifecycles - Modeling the World in States*. Prentice-Hall.

Stephen P.B. & A. Brad (2002). *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley.

Stojanovic, Z., A. Dahanayake & H. Sol (2001). A methodology framework for component-based system development support. *Proceedings of EMMSAD'01*, Interlaken, Switzerland, June 4-5.

Storey, V.C., R.H.L. Chiang, D. Dey, R.C. Goldstein & S. Sundaresan. (1997). Database design with common sense business reasoning and learning. *ACM Transactions Database Systems*, *22*(4), 471-512.

Skuce, D. & T. Lethbridge (1995). CODE4: A unified system for managing conceptual knowledge. *International Journal of Human-Computer Studies*, *42*, 413-451.

Turki, S. (2002). Hyperclasses: Towards a new kind of independence of the methods from the schema. *Proceedings of the 4th International Conference On Enterprise Information Systems*, Spain, April.

Wand, Y. & R. Weber. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems*, *3*, 217-237.

Wand, Y. & R. Weber. (2002). Information systems and conceptual modeling - A research agenda. *Information Systems Research*, *13*(4), 363-376.

Watson, I. (2003). *Applying Knowledge Management: Techniques for Building Corporate Memories*. Morgan Kaufmann Publishers.

Witt, G.C. & G.C. Simsion. (2004). *Data Modeling Essentials: Analysis, Design and Innovation* (3rd edition). Morgan Kaufmann Publishers.

XMI (2000). eXtensible metadata interchange (XMI) specification version 1.0, Document ad/00-06-01. Retrieved December 2, 2007, from http://www.omg.org.

SARBANES-OXLEY COMPLIANCE: NEW OPPORTUNITIES FOR INFORMATION TECHNOLOGY PROFESSIONALS

Gary P. Schneider, University of San Diego
Carol M. Bruton, California State University San Marcos

ABSTRACT

Much has been written in the business press and in academic journals about the Sarbanes-Oxley Act of 2002 (SOA) and how it will affect corporate governance and the practice of auditing and public accounting. Recent literature also discusses how the requirements of SOA might or might not better protect investors. Very little has been written that addresses how SOA will affect the duties and responsibilities of information technology (IT) professionals. This paper outlines the opportunities for IT professionals in designing the systems that will enable companies to comply with the SOA. The paper also contrasts the qualifications of IT professionals with respect to SOA compliance work with those of public accounting firm staff members.

INTRODUCTION

The Sarbanes-Oxley Act of 2002 (SOA) was passed in the United States (U.S. Code, 2002) in response to a series of significant failures in corporate governance, including Enron (Schwartz, 2001) and the related failure of accounting firm Arthur Andersen (Eichenwald, 2002), HealthSouth (Day, 2003), Tyco (Sorkin, 2002), and WorldCom (Moules and Larsen, 2003). Even Europeans, many of whom were convinced that this rash of management frauds were a result of American's hyper-capitalism mania and could never happen in the refined atmosphere of the continent, found that they were not immune when Parmalat's $15 billion in understated debt and huge overstatements of sales and earnings were exposed (Adams, 2003).

The SOA imposes a number of requirements on companies, their managers, and their directors. It also imposes a number of requirements on the systems of internal control used in companies. In the next section, we outline the requirements imposed by the SOA. In the section following that, we outline the specific impacts that the law will have on the job duties and responsibilities of IT professionals.

REQUIREMENTS OF THE SOA

The SOA includes 11 Titles (USC, 2002). Title I establishes the Public Company Accounting Oversight Board. Title II defines auditor independence. Title III discusses corporate responsibility. Title IV discusses enhanced financial disclosures. Title V discusses securities analyst conflicts of interest. Title VI discusses Securities and Exchange Commission (SEC) resources and authority. Title VII discusses the studies and reports that must be completed. Title VIII discusses corporate and criminal fraud accountability. Title IX discusses white-collar crime penalty enhancements. Title X discusses corporate tax returns. Title XI discusses corporate fraud and accountability. In this section, we review each relevant SOA Title to provide background for the rest of the paper.

The Public Company Accounting Oversight Board

Title I of the SOA creates the Public Company Accounting Oversight Board (PCAOB). The PCAOB is a new body that will oversee audits of publicly-held companies. The board is composed of five full-time independent members, only two of whom can be CPAs. Board members can serve up to two-five year terms. The PCAOB must submit an audited annual report to the SEC. Any public accounting firm wishing to audit public companies must register with the PCAOB. The registered public accounting firm must submit an annual report to the PCAOB including a list of the firms they audited in the past year and the fees received by the firm for audit services, other accounting services, and non-audit services. Each registered public accounting firm must pay an annual fee to the board to recover the cost of processing and reviewing applications and annual reports.

Title I gives the PCAOB authority to establish auditing, quality control, and ethical standards. The public accounting firms are required to have audit working papers and a second partner review. They must describe in the audit report the scope of the auditors testing of internal control structure and procedures of the company. The current audit standards for testing and reporting on internal control (as contained in Statements on Auditing Standards No. 55, No. 78, and No. 94, AICPA, 2003) still apply under the SOA. The board is required to conduct inspections of the registered public accounting firms to determine their compliance with the SOA. The inspection must be annual if the firm provides audit reports for more than 100 companies. If the firm provides audit reports for less than 100 companies the inspections are every three years. The PCAOB has the right to impose sanctions on registered firms including suspension or permanent revocation of the firm's registration. The SOA also applies to foreign public accounting firms that prepare audit reports for registered companies.

Independence of Auditors

Title II of the SOA deals with auditor independence. SOA reiterates the long-standing requirement that a public accounting firm cannot provide an audit client with bookkeeping services

and financial information systems services. The SOA goes on to preclude the provision of eight specific types of non-audit services, including appraisal and valuation services, actuarial services, internal audit services, management functions, human resource consulting services, investment adviser services (including broker and dealer services), and legal services. The SOA also gives the PCAOB a catch-all right to prohibit other services in the future as it deems necessary or appropriate. Many of these now-prohibited services provided large portions of public accounting firm revenues in recent years.

The company's audit committee must approve any services provided by the public accounting firm, including any tax work, as well as any other services. Any services the audit committee approves and the audit firm provides must be disclosed to investors.

Title II requires a rotation of audit partners every five years. It also requires that the auditor report to the audit committee rather than company management. The auditor must inform the audit committee of the accounting policies used by the client and must disclose all accounting treatments discussed with management. The auditor must provide the audit committee with any other material written communication between the auditor and any client personnel.

An employee of the audit firm cannot, upon leaving the firm, accept a position with a client firm in the capacity of chief executive officer, controller, chief financial officer, chief accounting officer, or any other similar position within 12 months of ending employment with the audit firm.

Corporate Responsibility and the Role of the Audit Committee

The SOA's Title III gives the audit committee full and unencumbered responsibility for the appointment, compensation, and oversight of the work of the audit firm. The members of the audit committee must be independent of the company. They cannot be employees or otherwise accept any consulting, advisory, or other compensatory fee from the company.

Title III requires the audit committee to establish procedures for the receipt and treatment of complaints received by the company regarding accounting, internal controls, or auditing matters. There needs to be a confidential and anonymous process within the company for submitting issues, concerns, and information to the audit committee.

The chief executive officer and chief financial offer must sign the SEC reports indicating that they have reviewed the report. These officers must certify that, based on the officer's knowledge, the report does not contain any untrue statement of a material fact and does not omit the statement of any material fact. The signing officers are responsible for establishing and maintaining internal controls and for reviewing the controls' effectiveness within 90 days of the date of the SEC report. The signing officers must report all significant deficiencies in internal control to the audit firm and to the audit committee and must report any fraud, whether or not the amount is material, that involves management or other employees who play a significant role in the design, operation, or evaluation of the company's internal controls.

This section also includes the specification of responsibilities for attorneys. One rule in this section requires attorneys to report evidence of any material violation of securities law or breach of fiduciary duty by the company or its agents to the chief legal counsel or the chief executive officer of the company. If the chief legal counsel or chief executive officer do not appropriately respond the attorney must then report the evidence directly to the audit committee.

Financial Disclosures

The SOA's Title IV requires disclosure of all off-balance sheet transactions and obligations, including contingent obligations, that might have a material current or future effect on financial condition. Title IV requires that the company monitor and review the amount of off-balance sheet transactions and the use of any special purpose entities. Pro forma information must be reconciled with generally accepted accounting principles and must not contain an untrue statement of material fact. Many types of executive loans, which have been prevalent in recent years, are curtailed under provisions of Title IV. The SOA requires that companies have a code of ethics for senior financial officers.

The financial disclosure provisions also contain a requirement that the annual report include a report on internal control. The report must state that internal control is the responsibility of management and must contain an assessment of the effectiveness of the internal control structure and procedures. The auditors must attest to, and report on, management's assessment. This does not mitigate the directors' role, however. The company must disclose whether or not at least one member of the audit committee is a financial expert. If one member is not a financial expert they must explain the reason.

Conflicts of Interest of Securities Analysts

Title V of the SOA provides that rules must be enacted where appropriate to address conflicts of interest that can arise when securities analysts recommend equity securities in research reports and public appearances. The goal of this SOA section is to improve the objectivity of investment research and provide investors with more reliable information.

Commission Resources and Authority

Title VI of the SOA discusses a need for increased resources for the SEC to carry out their duties. Many observers have been critical of the government's unwillingness to devote sufficient resources to SEC enforcement units. As the scandals that led to the SOA were unfolding, the SEC claimed it was understaffed. Since the SOA was enacted, significant increases in SEC enforcement budgets have not been forthcoming.

Studies and Reports

Title VII calls for a number of research studies to be conducted. One study would include research regarding the factors that led to the consolidation of public accounting. This consolidation has reduced the number of different and distinct firms capable of providing auditing services to large publicly-held companies. Another study is required that will investigate the role and function of credit rating agencies in the operation of the securities market. The SEC will conduct a study on the number of securities professionals who have been found to have aided and abetted a violation of Federal securities laws. The SEC will review and analyze each of its enforcement actions that involve violations of reporting requirements imposed under the securities laws and restatements of financial statements. The Comptroller General of the United States will conduct a study on whether investment banks and financial advisers helped companies manipulate their earnings with a goal of hiding the companies' true financial conditions.

Criminal Fraud and White Collar Crime Penalties

Titles VIII, IX, and XI include new definitions of criminal acts and provide a variety of new penalties and some increased penalties for existing crimes. Title VIII of the SOA provides penalties for destruction, alteration, or falsification of records in federal investigations and bankruptcy proceedings. It also prohibits and provides penalties for destruction of corporate audit records. This section calls for a review of Federal sentencing guidelines for obstruction of justice and criminal fraud convictions. It also provides whistleblower protection for employees of publicly traded companies. Specific enhanced criminal penalties are imposed for the act of defrauding the shareholders of publicly traded companies. Title IX increases the penalties for white-collar crime including fines and prison sentences.

Title XI provides new penalties for tampering with a record or impeding an official proceeding. It also increases the authority of the SEC to prohibit persons from serving as officers or directors. It provides specific fines and imprisonment terms for persons or organizations engaged in retaliation against informants.

Corporate Tax Returns

Title X states the opinion of the Senate that it would like to require that corporate federal income tax returns be signed by the chief executive officer of the filing entity. This is an additional indicator of the degree of responsibility viewed by the drafters of the SOA to be a necessary condition in the person of the chief executive officer. The SOA includes several signing requirements that many critics believe to be ceremonial and unsubstantial. However, many other observers note that a signed document is far more difficult to deny and that the signature

requirements could lessen the weight of a defense based on the chief executive officer not knowing what was happening in the company.

ROLE OF THE ACCOUNTING INDUSTRY IN SOA COMPLIANCE

The accounting industry has reacted rapidly to the passage of the SOA (AICPA, 2002a; AICPA, 2002b, AICPA, 2002c). Its reactions have been largely defensive. Many observers believe the accounting industry at least partially responsible for not detecting many of the recent frauds and accounting irregularities (Rezaee, 2003; Velayutham, 2003). Indeed, it is interesting to note that since 2002, when many news stories began reporting on these frauds and accounting failures, the news media has referred to "the accounting industry." In earlier years, the business was typically referred to as "the accounting profession." When the SOA was passed many accountants saw it as a combination of things. They saw it as an opportunity to repair their tarnished reputation, a chance for real reform, and even a way to replace lost consulting revenues with a new (and perfectly legal under the SOA) revenue stream: consulting services designed to help companies comply with the SOA (Munter, 2003). Needless to say, some accounting industry critics found this turn of events ironic.

Recent History of the Industry

The recent history of the accounting industry is interesting. As the market for audit services became increasingly competitive in the 1980s, firms attempted to contain costs and defend against litigation from users of financial statements that the firms had audited. To do this, accounting firms have increasingly lobbied for precise, mechanical accounting rules and have implemented standardized operating procedures. The goal was to reduce variability in the performance of audit work. Variations in audit work were perceived as costly and as opening the door for zealous plaintiffs to confuse juries and judges about the quality of the audit work performed (Healy, 2003). Because these heavily-lobbied regulators were pressed to create rules and legislation that would cover, specifically, all contingencies, accounting and audit standards have become incredibly detailed. Healy (2003) notes that the 2,300 pages of Financial Accounting Standards Board (FASB) standards that existed in 1985 had increased to 4,000 pages in 2002. This reliance on detailed rules and regulations has, as the Enron case illustrated so spectacularly (Schwartz, 2001), encouraged companies to enter business arrangements that satisfy the terms of the detailed rules, but that completely circumvent the intent of those rules.

A Trend Toward Standardization

Healy (2003) notes that a major problem with a standardized, rule-based auditing approach is that it gives audit firms a way to avoid judgment of the overall compliance of a auditee's financial

statements with generally accepted accounting principles. In effect, the pre-1980s auditor would subject the financial disclosures to an overall "smell test." If the detailed rules were followed, but the overall presentation was misleading, a company in the old days could count on a hard face-to-face meeting with the audit partner (Zeff, 2003). Healy (2003) notes that Arthur Andersen, in its audit of Enron's special purpose entities, determined that the company had satisfied all of the detailed rules for off-balance-sheet financing, but did not report to Enron's investors that the financial statements did not represent its true financial position. These entities met the requirements of the detailed rules, but flaunted the overall spirit of "fair presentation."

Decline in Quality of Inputs

In public accounting, a firm is only as good as the professional staff that work for the firm (Zeff, 2003). Partners in the 1960s used to describe their business as buying people by the year and renting them out by the hour. The inputs in the accounting business are the people in the business, particularly the new hires who perform most of the on-site audit work at client locations. Healy (2003) noted that the end of accounting as a "profession" probably occurred because the industry was no longer able to attract the best and brightest students graduating from college. Since the 1980s, fewer graduates with accounting majors have entered public accounting. The effect is particularly marked at top business schools. Healy (2003) reports that only three percent of Wharton's accounting graduates entered public accounting in 2002.

Need to Please Clients

Since the 1980s, audits have been viewed increasingly as a commodity service. One audit firm is as good as another, and no client really cares if they received a quality audit as long as they received the auditor's unqualified opinion (Zeff, 2003). This perception of audit services as a commodity lead to severe price competition (Healy, 2003). Accounting firms responded by offering a variety of consulting services. These services had higher margins then audit work and could be sold to audit clients. As clients provided more and more consulting revenues to their audit firms, the objectivity and independence of auditors came into question (Briloff, 1987; Stevens, 1991).

By the beginning of the 1980s, the large accounting firms had all concluded that profit margins on audits would be painfully thin, particularly relative to those on other financial services (Stevens, 1991). Their response was to diversify into other businesses- notably consulting (Zeff, 2003). Since audit quality did not matter to clients, auditors became more and more desperate to curry clients' favor by maintaining friendly relationships with client accounting managers and top executives so that the firm could bid on more and more lucrative consulting work with the client. Client retention and expansion of non-audit fee revenue became important parts of accounting firm employees' compensation arrangements. For partners in the firms, it was a critical element (Healy, 2003; Zeff, 2003).

Ability of Accounting Firms to Provide SOA Assistance

Clearly, accountants and public accounting firms have the technical skills to provide help to companies that need assistance in complying with SOA (Coustan, et al., 2004; Winters, 2004). Lanza (2004) suggests that company's internal audit staff might be valuable consultants for SOA compliance and systems design and development. Indeed, many current textbooks for the accounting information systems course, which is required of accounting majors at most universities, include detailed coverage of internal controls, internal control assessment techniques, and current applications of information technology to the tasks needed to comply with SOA (see, for example, Gelinas and Sutton, 2002; Hall, 2004; or Romney and Steinbart, 2002). Despite these arguments for technical competence, the decline in the quality of recruits to accounting firms and the public accounting industry's recent failure to show itself to be a deserving recipient of companies' (and the public's) trust, we argue that many of the important elements required by SOA might be best addressed by using the consulting expertise of IT professionals.

IT PROFESSIONALS AND THE DEMANDS OF THE SOA

An understanding of internal control demands an understanding of the underlying accounting and administrative systems of the company (Hall, 2004). As every business of any size has computerized its accounting and administrative systems, the people who know these systems well and who understand their design are increasingly members of the ranks of IT professionals. In this section, we argue that IT professionals, both inside the company and in consulting firms outside the company, can provide valuable services to the company as it attempts to comply with the internal control standards set by the SOA. Further, the IT professionals who have gone on to become lawyers practicing in the area of high technology are especially well-qualified to offer SOA consulting services because of their unique combination of IT knowledge and legal training.

Technical Skills and Business Knowledge of IT Professionals

IT professionals have been engaged in the design and implementation of systems for decades, far longer than accountants have been seriously involved in these issues (Gelinas and Sutton, 2002). They have a keen understanding of what it takes to make these systems work. Increasingly, IT professionals are educated, trained, and respected as business analysts as well as for their technical knowledge.

Lanza (2004) notes that two of the most important elements of any SOA compliance program is the proper use of data analysis tools and data mining software. Data analysis functions include the use of query tools that allow users to ask questions of the enterprise-wide information system (Gelinas, 2002). In large organizations such as those subject to SOA, this system will, in most cases, have been designed and implemented by the company's IT staff. It will definitely be maintained by

IT staff. The people who know the most about the enterprise-wide information system will always be IT professionals. Many companies have undertaken major knowledge management initiatives in recent years (Angus, 2003; Awad and Ghaziri, 2003). These initiatives have, in most cases, been designed and implemented by IT professionals. As SOA requirements become part of the fabric of large companies, they will be included as part of these companies' knowledge management systems (Lanza, 2004).

Winters (2004) outlines three questions that an SOA consultant should be able to help a company answer: (1) is it better to develop a short term solution or a more sustainable one for the long term, (2) which software tools are best able to provide complete, effective, and sustainable compliance, and (3) what other policies, training programs, and other investments are needed to comply with SOA and maximize the utilization of the software in the context of the company's existing information systems. We argue that IT professionals would provide better advice regarding each of these three questions given the skill sets and business knowledge generally agreed to reside in IT staff (Laudon and Laudon, 2004; McLeod and Schell, 2004; Oz, 2004).

Independence of IT Professionals

Although IT professionals employed by the company are not, by definition, independent, they often operate with considerable latitude. Because IT professionals have a level of expertise that can be critical to company operations, they often can derive a level of mystique that provides a level of independence (Burns and Haga, 1977). External consultants that offer companies IT advice are likely to be much more independent than public accounting firms and they are not tarnished by association with the very evils that prompted the legislation.

CONCLUSION

We have examined the requirements of the SOA and what companies must do to their accounting and internal control systems to comply with the law. After considering accountants in the company and external public accounting firms as likely candidates for the job of advising companies what they must do to comply with the SOA, we find them lacking in the key elements of technical expertise, independence, and overall business knowledge. We argue that IT professionals have higher degrees of relevant technical expertise and sufficient levels of overall business knowledge to be very qualified to advise companies on SOA compliance efforts, especially if their technical knowledge is augmented by legal training. This legal training is relevant to some areas of SOA compliance than others. In the final analysis, IT professionals have a strong advantage over the accounting industry in this comparison: IT professionals are not tarred by an association with the frauds, irregularities, and crimes that motivated the SOA's passage. Accountants in general and public accounting firms in particular, cannot make that claim.

REFERENCES

American Institute of Certified Public Accountants (AICPA) (2002a). How the Sarbanes-Oxley Act of 2002 impacts the accounting profession, AICPA Web site. Retrieved August 13, 2003, from http://www.aicpa.org/info/Sarbanes-Oxley2002.asp

American Institute of Certified Public Accountants (AICPA) (2002b). Landmark accounting reform legislation signed into law, *CPA Letter*. Retrieved August 19, 2003, from http://www.aicpa.org/pubs/cpaltr/Sept2002/landmark.htm

American Institute of Certified Public Accountants (AICPA). 2002c. Additional aspects of Sarbanes-Oxley Act explained, *CPA Letter*. Retrieved August 20, 2003, from http://www.aicpa.org/pubs/cpaltr/Oct2002/add.htm

American Institute of Certified Public Accountants (AICPA) (2003). *AICPA Professional Standards*. New York: AICPA.

Angus, J. (2003). Rethinking knowledge management. *InfoWorld*, 25(17), March 17, 32-35.

Awad, E. & H. Ghaziri (2003). *Knowledge management*. Upper Saddle River, NJ: Prentice-Hall.

Briloff. A. (1987). Do management services endanger independence and objectivity? *The CPA Journal*, 57(8), August, 22-29.

Burns D. & W. Haga (1977). Much ado about professionalism: A second look at accounting. *Accounting Review*, 52(3), July, 705-715.

Coustan, H., L. Leinicke, W. Rexroad & J. Ostrosky (2004). Sarbanes-Oxley: What it means to the marketplace. *Journal of Accountancy*, 197(2), February, 43-47.

Day, K. (2003). SEC sues HealthSouth, CEO over earnings: Former CEO pleads guilty to fraud, *The Washington Post*, March 20, E1.

Eichenwald, K. (2002). Andersen guilty in effort to block inquiry on Enron, *The New York Times*, June 16, 1.

Gelinas, U. & S. Sutton (2002). *Accounting information systems*, (5th ed.) Cincinnati: South-Western..

Hall, J. (2004). *Accounting information systems*, (4th ed.) Cincinnati: South-Western.

Hardesty, D. (2004). *Practical guide to corporate governance and accounting: Implementing the requirements of the Sarbanes-Oxley Act*. Boston: Warren, Gorham & Lamont.

Healy, P. (2003). How the quest for efficiency corroded the market, *Harvard Business Review*, 81(7), July,

Lanza, R. (2004). Making sense of Sarbanes-Oxley tools, *Internal Auditor*, 61(1), February, 45-49.

Laudon, K, & J. Laudon (2004). *Management information systems*, (8th ed.) Upper Saddle River, NJ: Prentice-Hall.

McLeod, R. & G. Schell (2004). *Management information systems*, (9th ed.). Upper Saddle River, NJ: Prentice-Hall.

Moules, J.& P. Larsen (2003). Reports condemn culture of fraud at WorldCom, *Financial Times*, June 10, 1.

Munter, P. (2003). Evaluating internal controls and auditor independence under Sarbanes-Oxley. *Financial Executive*, 19(7), October, 26-27.

Oz, E. (2004). *Management information systems*, (4th ed.) Boston: Course Technology.

Rezaee, Z. (2003). Restoring public trust in the accounting profession by developing anti-fraud education, programs, and auditing, *Managerial Auditing Journal*, 19(1), 134-148.

Romney, M. & P. Steinbart (2002). *Accounting information systems*, (9th ed.) Upper Saddle River, NJ: Prentice-Hall.

Schwartz, N. (2001). Enron fallout: Wide, but not deep, *Fortune*, 144(13), December 24, 71-72.

Sorkin, A. (2002). Tyco figure pays $22.5 million in guilt plea, *The New York Times*, December 18, 1.

Stevens, M. (1991). *The big six: The selling out of America's top accounting firms*. New York: Simon & Schuster.

United States Code (2002). *Sarbanes-Oxley Act of 2002*, Public Law No. 107-204, codified at 15 U.S.C. §7201

Velayutham, S. (2003). The accounting profession's code of ethics: Is it a code of ethics or a code of quality assurance? *Critical Perspectives on Accounting*, 14(4), May, 483-503.

Winters, B. (2004). Choose the right tools for internal control reporting, *Journal of Accountancy*, 197(2), February, 34-40.

Zeff, S. (2003). How the U.S. accounting profession got where it is today: Part II, *Accounting Horizons*, 17(4), December, 267-286.

REASSESSMENT OF THE RELATIVE WEIGHTS OF GPA AND GMAT IN THE MBA ADMISSION DECISION PROCESS

Constantine Loucopoulos, Northeastern Illinois University
Charletta F. Gutierrez, Northeastern Illinois University
Durward Hofler, Northeastern Illinois University

ABSTRACT

The explosive growth of MBA programs has generated the need to streamline the admission process, by assigning the appropriate weights to admission criteria like undergraduate GPA and GMAT that accurately reflect each school's prior admission decisions. A mathematical programming model is proposed for the assignment of relative weights to undergraduate GPA and GMAT in the MBA admission process. The proposed model classifies applicants as admissible, borderline and non-admissible with the objective of minimizing the number of applicants with mis-specified admissibility status. The model can easily be extended to account for other variables that may affect an applicant's admissibility.

INTRODUCTION

Graduate schools of business use a variety of criteria to assess the admissibility of applicants to their MBA programs. Usually such criteria include the overall undergraduate *GPA* (or the *GPA* for the last 60 credit hours of undergraduate studies) and the *GMAT* score. According to the Graduate Management Admission Council (2006) that sponsors the Graduate Management Admission Test, "*GMAT* scores are used by more than 3,000 graduate management programs from more than 1,500 graduate management schools, many of which require *GMAT* scores from each applicant." A common practice of a large number of graduate schools of business has been to use the composite score *200*GPA + GMAT* to establish the admissibility of applicants to MBA programs. However, does such composite accurately reflect the weights assigned by different academic institutions to these two admission criteria?

The use of expert systems (Finlay & King, 1989; Moore, 1998), neural networks (Hardgrave, Wilson & Walstrom, 1994; Naik & Ragothaman, 2004), regression analysis (Wright & Palmer, 1994; Braunstein, 2002), and discriminant analysis (Wilson & Hardgrave, 1995) have been proposed for determining the admissibility of applicants to MBA programs. Hardgrave, Wilson and Walstrom (1994, p. 251) point out that "Categorization techniques such as discriminant analysis, would appear

to be more appropriate than regression analysis for predicting MBA success, since this method could predict MBA applicants into one category, such as successful/not successful, rather than trying to predict a specific *GPA*."

If discriminant analysis is to be used for the determination of the admissibility status of applicants to an MBA program, the question that arises is whether a mathematical programming-based classification procedure should be used instead of one of the standard parametric classification techniques. Mathematical programming approaches to the statistical classification problem have received considerable research interest in the last 25 years. An advantage of using mathematical programming instead of the standard parametric classification techniques like Fisher's (1936) linear discriminant function or Smith's (1947) quadratic discriminant function, is that such mathematical programming models do not make any assumptions about the covariance structures or about multivariate normality for optimal classificatory performance. Various studies have shown that mathematical programming models can outperform the standard parametric classification procedures for a variety of data configurations (Stam & Jones, 1990; Gochet, Stam, Srinivasan & Chen, 1997; Loucopoulos & Pavur, 1997).

PROPOSED MODEL

In this paper, we propose a mathematical programming model for the assignment of the appropriate relative weights to undergraduate *GPA* and *GMAT* scores in the MBA admission decision process. The proposed model minimizes the number of applicants who are erroneously classified in terms of their admissibility status. The proposed model generates a composite score $a_0 + a_1 * GPA + a_2 * GMAT$ for each applicant. An applicant should be admitted if the composite score is at least as large as a preassigned value, e.g. s_1. Another preassigned value in the model is g, which is equal to 10^{-p} with p denoting the number of decimal digits in the reported value of the composite score $a_0 + a_1 * GPA + a_2 * GMAT$. If the composite score is between 0 and s_1-g, then the applicant should be classified as borderline. Correspondingly, if the composite score is less than or equal to -g, then the applicant should not be admitted into the MBA program. The objective of the model is to minimize the number of applicants with mis-specified admissibility status.

Notation:

a_1	is the weight assigned to the undergraduate *GPA*
$X_1^{(i)}$	is the undergraduate *GPA* of applicant i ($i = 1, 2, ..., n$)
a_2	is the weight assigned to the *GMAT* score
$X_2^{(i)}$	is the *GMAT* score of applicant i ($i = 1, 2, ..., n$)

a_0 is a locational adjustment to the composite score

n is the number of applicants

s_1 is a preassigned constant equal to the lowest composite score for an applicant to be admissible

$$I_i = \begin{cases} 1 & \text{if the admissibility status of the applicant } i \text{ has been misspecified} \\ 0 & \text{otherwise} \end{cases}$$

g is a preassigned constant equal to 10^{-p} with p denoting the number of decimal digits in the reported value of the composite score

M is a preassigned constant that is multiplied by the binary variable I_1 in order to account for incorrectly classified applicants

Formulation:

$$\text{min Sum } I_i \text{ for } i = 1 \text{ to } n \qquad \text{Formula (1)}$$

subject to

$$a_0 + a_1X_1^{(i)} + a_2X_2^{(i)} + MI_i \geq s_1 \quad \text{for all admissible applicants } i \qquad \text{Formula (2)}$$

$$a_0 + a_1X_1^{(i)} + a_2X_2^{(i)} - MI_i \leq s_1 - g \quad \text{for all borderline applicants } i \qquad \text{Formula (3)}$$

$$a_0 + a_1X_1^{(i)} + a_2X_2^{(i)} + MI_i \geq 0 \quad \text{for all borderline applicants } i \qquad \text{Formula (4)}$$

$$a_0 + a_1X_1^{(i)} + a_2X_2^{(i)} - MI_i \leq -g \quad \text{for all non-admissible applicants } i \qquad \text{Formula (5)}$$

where a_0, a_1 and a_2 are sign-unrestricted variables. It is expected that both a_1 and a_2 will be positive and a_0 will be negative at the optimal solution.

The value assigned to M should be substantially higher than s_1, the lowest score needed for admission. Thus, if the lowest score needed for admission is set equal to *50*, then M could be *100*, *200* or *250*. Koehler and Erenguc (1990) note that in mathematical programming approaches to the classification problem "We rely on the standard maxim in integer programming to choose M large enough…" According to Stam and Jones (1990), "The parameter M is an arbitrary, sufficiently

large real-valued constant." The numerical value assigned to *M* does not affect the performance of the model, as long as it is substantially larger than the value of s_1. Such value of M guarantees that an applicant may be classified to any one of the three groups. The model determines the weights assigned to the *GPA* and *GMAT* that will yield the minimum number of applicants with mis-specified admissibility status.

It should be noted that an academic institution may use this model with additional variables. In such case, the composite score $a_0 + a_1X_1^{(i)} + a_2X_2^{(i)}$ will be replaced with

$$a_0 + Sum\ a_kX_k^{(i)}\ for\ k = 1\ to\ r \qquad Formula\ (6)$$

where *r* is the number of variables used for the computation of the composite score. Thus, for example, an academic institution may use a quantitative measure of an applicant's computer literacy in the decision process for admission to its MBA program.

PERFORMANCE OF MODEL

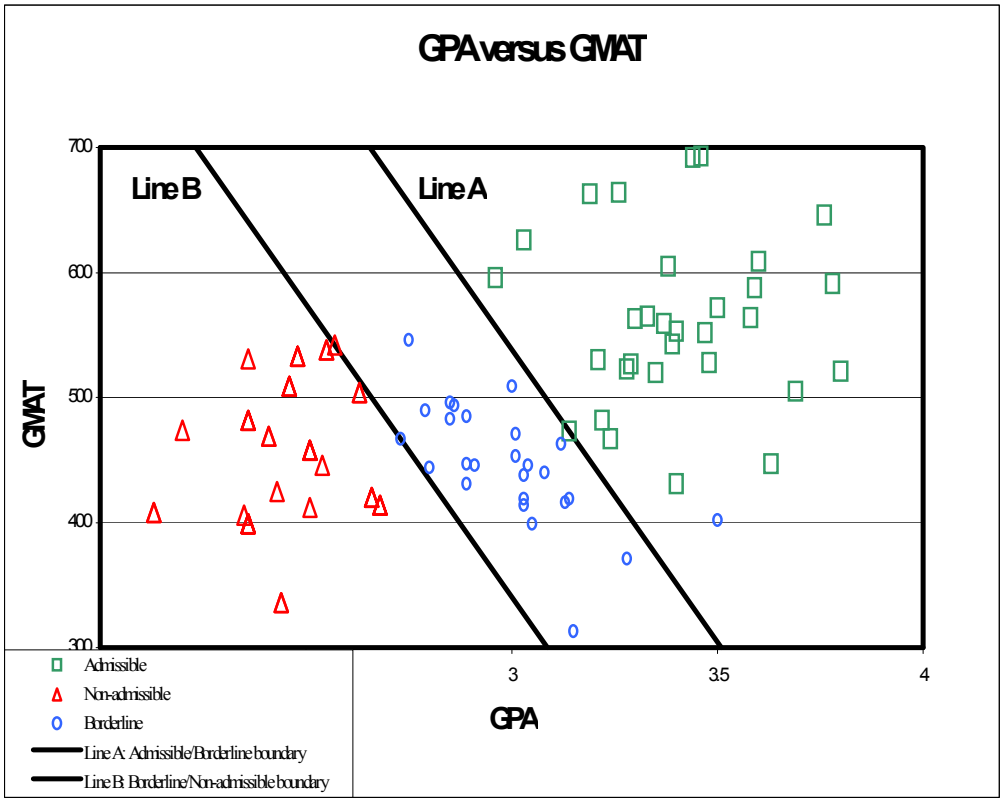
The performance of the proposed model was tested using the MBA admission data in Johnson and Wichern (2002). Eight-five applicants to an MBA program were classified as admissible, borderline, or non-admissible based on their undergraduate *GPA* and *GMAT* scores. Out of the *85* applicants, *31* were admissible, *26* were borderline, and *28* were non-admissible. The eventual admissibility status of the *26* borderline applicants would be determined by the inclusion of other criteria, like letters of recommendations from professors and employers, record of undergraduate campus activities, applicant's statement of professional goals, or previous work experience. Using the proposed model with $M = 100$, $s_1 = 50$ (lowest composite score for an applicant to be admissible) and $g = .0001$, the model was successful in classifying correctly *84* out of the *85* applicants. One borderline applicant was classified as admissible by the model. In order to assess if the classification accuracy of the proposed model adds value to the MBA admission decision process, Huberty's (1984) standardized normal test statistic was used:

$$z = (n_{gg} - e_g)\ SQRT(n_g)\ /SQRT(e_g(n_g - e_g)) \qquad Formula\ (7)$$

In this statistic, n_g is the actual number of observations from group G_g, e_g is the number of correctly classified observations from G_g expected by chance, and n_{gg} is the actual number of observations from group G_g that are correctly classified. The values of the test statistic for the admissible, borderline and non-admissible groups are *7.348, 7.256* and *7.550*, respectively, with $p < .001$ in each of the three cases. It should be noted that, if all the observations from one group are correctly classified (as was the case for the admissible and non-admissible groups), then the value of the test statistic for that group is $SQRT(n - n_g)$.

The derived composite formula was *-440.6649 + 118.25924 * GPA + .2522851 * GMAT.* Alternatively, an applicant is admissible if *468.7524 * GPA + GMAT ≥ 1944.8826*. An applicant is non-admissible if *468.7514 * GPA + GMAT ≤ 1746.6937*. These classification results are shown in Figure I, where the equation of Line A is *468.7524*GPA + GMAT = 1944.8826* and the equation of Line B is *468.7514 * GPA + GMAT = 1746.6937*. In order to make these formulas "user-friendly," an academic institution may elect to use *470 * GPA + GMAT ≥ 1950* for an applicant to be admissible, and *470 * GPA + GMAT ≤ 1750* for an applicant to be non-admissible. In such case, the number of incorrectly classified applicants would increase from one to two. One admissible applicant would be classified as borderline, and one borderline applicant would be classified as admissible.

Figure 1



Suppose that the same academic institution was using the composite score *200*GPA + GMAT*. Then the composite scores of the admitted applicants would range from *1101* to *1398*, the scores of the borderline applicants would be between *943* and 1109, whereas the scores of the non-admissible applicants would range from *791* to *1056*. With such ranges of composite scores for the

three groups of applicants, what would be the optimal cutoff composite scores that would minimize the number of erroneously classified applicants? This is obtained by setting the range *1004* to *1110* as borderline. As a result, one admitted applicant should have been classified as borderline (score of *1101*), five non-admissible applicants should have been classified as borderline (scores of *1056*, *1048*, *1038*, *1030* and *1029*), whereas one borderline applicant should have been classified as non-admissible (score of *943*). Thus, using the composite *200*GPA + GMAT*, the minimum number of incorrectly classified applicants is seven. Thus, by assigning a relative weight of *468.7524* to the *GPA*, instead of *200*, the number of incorrectly classified applicants is reduced from seven to one.

The classificatory performance of the proposed model was also compared against performance of the standard parametric classification procedures. Fisher's linear discriminant function (LDF) resulted in seven misclassified applicants, whereas Smith's quadratic discriminant function (QDF) resulted in three misclassified applicants. These results are obtained when the entire sample of 85 applicants is used as a training sample. Furthermore, the performance of the proposed model was compared against that of LDF and QDF using Lachenbruch's (1967) leave-one-out cross-validation procedure. Using this procedure, an observation is left out and classification functions are constructed using the remaining *n-1* observations. Then these classification functions are used to classify the left out observation into one of the groups. This process is equivalent to using a training sample of size *n-1* and validation sample of *1*, and is repeated n times. Thus, each time *84* applicants will be used to construct the classification function and the left out applicant will be used for validation purposes, with the process being repeated *85* times (once for each of the *85* applicants). Thus, an assessment can be made as to how well the proposed model performs in classifying new applicants. Using the leave-one-out cross-validation, there were two students misclassified by the proposed model. LDF yielded nine misclassified applicants, whereas QDF misclassified four applicants.

CONCLUSION

In this paper, we propose a mathematical programming model for the determination of the relative weights to be assigned to the undergraduate *GPA* and *GMAT* scores in the MBA admission decision process. This model gives academic institutions the opportunity to assess the relative weights of undergraduate *GPA* and *GMAT* that accurately reflect their prior admission decisions, and assists such institutions in streamlining their admission decision processes. It is shown that the proposed mathematical programming model for the determination of admissibility status exhibits higher classificatory performance than the standard parametric classification procedures. It should be noted that the weights assigned to *GPA* and *GMAT* using the proposed model are data dependent, and thus will differ from one academic institution to another.

REFERENCES

Braunstein, A. W. (2002). Factors determining success in a graduate Business program. *College Student Journal*, *36*, 471-477.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179-188.

Gochet, W., A. Stam, V. Srinivasan & S. Chen (1997). Multigroup discriminant analysis using linear programming. *Operations Research*, *45*, 213-225.

Graduate Management Admission Council (2005). *GMAT Information Bulletin for Testing in 2006*, McLean, VA:GMAC.

Hardgrave, B. C., R. L. Wilson & K. A. Walstrom (1994). Predicting graduate student success: A comparison of neural networks and traditional techniques. *Computers and Operations Research*, *21*, 249-263.

Huberty, C. J. (1984). Issues in the use and interpretation of discriminant analysis. *Psychological Bulletin, 95*, 156-171.

Johnson, R. A., & D. W. Wichern (2002). *Applied Multivariate Statistical Analysis (Fifth Edition).* Englewood Cliffs, NJ: Prentice Hall.

Koehler, G. J. & S. S. Erenguc (1990). Minimizing misclassifications in linear discriminant analysis. *Decision Sciences, 21*(1), 63-85.

Lachenbruch, P.A.(1967). An almost unbiased method of obtaining confidence intervals for the probability of misclassifications in discriminant analysis. *Biometrics*, *23*, 639-645.

Loucopoulos, C. & R. Pavur (1997). Examination of the classificatory performance of mathematical programming approaches to the three-group discriminant problem: The case of small samples. *Annals of Operations Research*, *74*, 191-209.

Naik, B. & S. Ragothaman (2004). Using neural networks to predict MBA student success. *College Student Journal*, *38*, 143-149.

Smith, C. A. B. (1947). Some examples of discrimination. *Annals of Eugenics*, *13*, 272-282.

Stam, A. & D. Jones (1990). Classification performance of mathematical programming techniques in discriminant analysis: Results for small and medium sample sizes. *Managerial and Decision Economics*, *11*, 243-253.

Wilson, R. L. & B. C. Hardgrave (1995). Predicting graduate student success in an MBA program: Regression versus classification. *Educational and Psychological Measurement*, *55*(2), 186-195.

Wright, R. E. & J. C. Palmer (1994). GMAT scores and undergraduate GPAs as predictors of performance in graduate business programs. *Journal of Education for Business*, *9*, 344-348.