Volume 9, No. 5
October 2011
ISSN: 1545-679X

INFORMATION SYSTEMS EDUCATION JOURNAL

In this issue:

EDSIG
Serving Information Systems Educators

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **EDSIG**, the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is quarterly. The first year of publication is 2003.

ISEDJ is published online (http://isedjorg) in connection with ISECON, the Information Systems Education Conference, which is also double-blind peer reviewed. Our sister publication, the Proceedings of ISECON (http://isecon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is about 45%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org.

2011 AITP Education Special Interest Group (EDSIG) Board of Directors

Alan Peslak
Penn State University
President 2011

Wendy Ceccucci
Quinnipiac University
Vice President

Tom Janicki
Univ of NC Wilmington
President 2009-2010

Scott Hunsinger
Appalachian State University
Membership Director

Michael Smith
High Point University
Secretary

Brenda McAleer
Univ of Maine Augusta
Treasurer

Michael Battig
Saint Michael's College
Director

George Nezlek
Grand Valley State University
Director

Leslie J. Waguespack Jr
Bentley University
Director

Mary Lind
North Carolina A&T St Univ
Director

Li-Jen Shannon
Sam Houston State Univ
Director

S. E. Kruck
James Madison University
JISE Editor

Kevin Jetton
Texas State University
FITE Liaison

Copyright © 2011 by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to Wendy Ceccucci, Editor, editor@isedj.org.

INFORMATION SYSTEMS
EDUCATION JOURNAL

Editors

Wendy Ceccucci
Senior Editor
Quinnipiac University

Thomas Janicki
Publisher
Univ NC Wilmington

Don Colton
Emeritus Editor
Brigham Young University
Hawaii

Nita Brooks
Associate Editor
Middle Tennessee
State University

Mike Smith
Associate Editor - Cases
High Point University

ISEDJ Editorial Board

Alan Abrahams
Virginia Tech

Mike Battig
Saint Michael's College

Gerald DeHondt II
Grand Valley State University

Janet Helwig
Dominican University

Mark Jones
Lock Haven University

Cynthia Martincic
Saint Vincent College

Brenda McAleer
University of Maine at Augusta

Monica Parzinger
St. Mary's University
San Antonio

Doncho Petkov
Eastern Connecticut State Univ.

Samuel Sambasivam
Azusa Pacific University

Mark Segall
Metropolitan State College of
Denver

Li-Jen Shannon
Sam Houston State University

Karthikeyan Umapathy
University of North Florida

Laurie Werner
Miami University

Bruce White
Quinnipiac University

Charles Woratschek
Robert Morris University.

Peter Y. Wu
Robert Morris University

©2011 EDSIG (Education Special Interest Group of the AITP)

Defining the Content of the Undergraduate Systems Analysis and Design Course as Measured by a Survey of Instructors

Timothy J. Burns
tburns1@ramapo.edu
Anisfield School of Business,
Ramapo College of New Jersey
Mahwah, New Jersey 07430, USA

Abstract

There are many factors that make the undergraduate systems analysis and design course somewhat enigmatic in its purpose, and therefore equivocal in its delivery. The purpose of this research is to learn, specifically, what instructors are teaching in their systems analysis and design courses. This paper reports the results of a survey and follow up interviews that were administered to instructors of the course located in colleges and universities around the world. Results indicate that there is a fair amount of consensus among instructors as to the course content.

Keywords: IS research toward educators, pedagogy, IS undergraduate curriculum, Teaching Systems Analysis and Design

1. INTRODUCTION

In their 2006 article introducing a special issue on teaching systems analysis and design (SA&D), Harris et al. made the statement that "If you were to assemble 50 IS professors into one room you would likely get 50 different opinions on how best to teach SA&D" (Harris, Lang, Oates, & Siau, 2006). The purpose of this research is to test that statement, but on a slightly larger scale. Over 1500 Information Systems (IS) professors were asked their opinion on how best to teach SA&D. Although only 172 chose to answer, it is felt that this is a large enough sample to get a good idea as to the heterogeneity (and conversely the homogeneity) of the course content of undergraduate SA&D courses across colleges and universities around the world. In essence, we want to know what instructors are teaching in their SA&D courses.

This article is an attempt to go further in depth and breadth than a similar article written in 2005. In that article, course materials were looked at to determine what textbooks and system development methodologies were being covered in system development courses (Burns & Klashner, 2005). This article goes further in breadth in that it looks at all materials covered in the courses. It goes further in depth in that it delves deeper into instructor backgrounds, teaching methods, topics covered, etc.

2. BACKGROUND

As instructors of an SA&D course, we are constantly asking ourselves how best to conduct the course. How can we deliver the best product to prepare and educate students for a career in information systems development? Gorgone et al. define the purpose of an IS curriculum to be to "produce graduates equipped to function in entry level information systems positions with a strong basis for continued career growth" (Gorgone, Davis, Valacich, Topi, Feinstein, & Longenecker, 2002). Gorgone et al. also provide a fairly comprehensive list of what should be covered by the IS curriculum and what skills and perspectives IS graduates should have (Gorgone et al., 2002). They say that IS graduates should have strong analytical, critical thinking, interpersonal, and team skills. Furthermore, they should have broad business and real world perspectives. Finally, they should have strong ethical principles and must be able to design and implement IT solutions that enhance organizational performance (Gorgone et al., 2002).

The SA&D course is an essential part of the IS curriculum (Harris et al., 2006). According to Gorgone et al., it should provide experience in determining system requirements and developing a logical design. Furthermore, students should work in teams to analyze problems and design and implement information systems (Gorgone et al., 2002).

These seem like clear cut goals, but there are many factors that make the SA&D course somewhat enigmatic in its purpose, and therefore equivocal in its delivery. A literature review of what SA&D instructors are researching, writing, and teaching in their SA&D courses shows that there are many issues that raise debate. There appear to be two predicating issues. First, is the issue of the volume of material that could be potentially covered in the course. SA&D encompasses a large and ever expanding field of material. How does the instructor choose what is important to cover? This raises a paradox between the volume of the material available to deliver and the amount of material that can be successfully delivered to the students. Many students get bored or overwhelmed with the material. Where on the spectrum between breadth and depth should the course content land?

The second predicating issue is the applied nature of the IS discipline and the ever changing dynamic of the field. Should we be teaching fundamental, tried and true concepts, methods, tools, and techniques, (many of which are 30 or more years old)? Or, should we be teaching the latest and most current of the concepts, methods, tools, and techniques being introduced in industry or by academics, even if they have not yet been completely vetted?

There are many other areas of contention when discussing the content of SA&D courses. For instance, currently there are two general overarching approaches to systems development; the traditional (or structured approach) and the object-oriented approach. Many instructors feel that one should be covered over the other (Rob 2006) or that perhaps the traditional approach is outdated. However, there is research that shows that teaching IS students the traditional approach is still a viable vehicle for SA&D instruction (van Vliet & Pietron, 2006).

Another area of contention surrounding the SA&D course is whether the content should be delivered in one course or two. Given the volume of material a two course model seems to makes sense. Many institutions have adopted this two course model. However, a two course paradigm may not fit in a curriculum where there are a limited number of IS specific course credits available. This is particularly true if the IS program is housed in a business program where many of the available credits are eaten up by core business courses.

One solution to fitting two SA&D courses into the curriculum is to make the second course part of the final capstone course in the program. Using this approach eases the burden of trying to deliver the large volume of course content, and it solves some other issues as well. In particular, it addresses the issue of where in the sequence of courses in an IS curriculum the SA&D course should lie. There are many who feel that even if the SA&D content is delivered as one course, it should be near the end of the IS course sequence. That is because the SA&D course draws upon the material presented in other IS core courses, such as database and network design, programming, project management, etc.

With two SA&D courses, one focused on basic theoretical foundations can be placed earlier in the course sequence, and one focused on the application of the knowledge gained can be placed near the end. The capstone course can be used to simulate a real world project where the students solve real business problems and work with real users.

This brings up yet another point of contention surrounding the SA&D course. Should the course include case studies and course projects that allow the students to apply their knowledge? Or should SA&D course content be delivered primarily through the traditional lecture? Studies have shown that traditional lecturing is by far the most common method for delivering course content and yet, is one of the least effective in terms of how much of the material that the students retain (Griffiths & Oates, 2003).

Introducing case studies and projects (simulated or real world) into the course has several advantages. Using case studies gives meaning and real world context to the material (Avison & Cole, 2006). Other techniques such as the use of "assumption/implication" debates, where the instructor states an assumption and then asks the students the implications, can provide a greater depth of understanding to the students while at the same time making the course material less boring (Avison et al., 2006).

3. RESEARCH METHODOLOGY

This research was conducted using a "grounded theory" approach. Grounded theory was developed by the sociologists Barney Glaser and Anselm Strauss in the 1960's. In the grounded theory approach, conclusions are drawn and theories are produced by analyzing a body of data. In essence, the theories that are produced are "grounded" in the data (Glaser & Strauss, 1967).

For this study, the process began by analyzing the current body of literature on teaching the SA&D course. This allowed the researchers to create a survey instrument that would be used to ask questions about the delivery of the SA&D course and the demographical background of the instructors and institutions that delivered those courses. The survey included mostly closed end questions (which are listed in the results section below) and a few open ended questions shown in the appendices at the end of the paper. The appendices show a summary of the responses to the open ended questions.

A list that contained approximately 1500 names of IS instructors was compiled and an email was then sent to every person on the list. The email explained the purpose of the study and provided a link that the subject could click on to complete the questionnaire. Approximately 172 people chose to participate in the study. Once the initial results had been tabulated, a follow up email was sent to all of the participants in order to gain a deeper understanding of their responses.

4. RESULTS

In this section, the data that was collected is summarized and presented as a series of tables. The survey questions are included to provide additional clarity.

Question 1: How do you determine what subjects and material to cover in your Systems Analysis & Design course? (Multiple Answers Allowed)

Table 1 How Instructors Choose their SA&D Course Content

Most often chosen combination:

Based on industry experience, feedback, or trends, the textbook, and academic literature

Based on industry experience, feedback, or trends	83%
Based on the textbook	70%
Based on academic literature	45%
Based on academic suggested course outline	24%
Mandated by college or department	12%
Other	13%

Question 2: What textbook(s) do you use in your course?

Table 2 Textbooks Used in SA&D Courses By Percentage of Respondents

Whitten & Bentley, "Systems Analysis and Design Methods"	18%
Dennis, Wixom, & Roth "Systems Analysis & Design"	12%
Shelly, Cashman, & Rosenblatt, "Systems Design & Analysis"	11%
Satzinger, Jackson, & Burd: "Systems Analysis and Design in a Changing World"	6%
Dennis, Wixom, & Tegarden: "Systems Analysis and Design with UML"	6%
Hofer, George, & Valcich: "Modern Systems Analysis and Design"	6%
Valacich, George, & Hoffer: "Essentials of Systems Analysis and Design"	6%
Own Material	5%
Whitten & Bentley: "Introduction to Systems Analysis and Design"	4%
Kendall & Kendall: "Systems	4%

Analysis and Design”	
George, Batra, Valacich, & Hoffer: “Object-Oriented Systems Analysis and Design”	4%
Shelly & Rosenblatt: “Systems Analysis & Design”	3%
Marakas: “Systems Analysis & Design: An Active Approach”	3%
DeWitz: “Systems Analysis and Design and the Transition to Objects**”**	2%
Harris, “Systems Analysis and Design for the Small Enterprise”	2%
Larman: “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”	2%
Other	6%

Question 3: How did you determine what textbook(s) to use in your Systems Analysis & Design course? (Multiple Answers Allowed)

Table 3 How Respondents Determined What Textbook to Use

Based on what I feel the course should cover	73%
Based on industry experience, feedback, or trends	39%
Suggested by a colleague	19%
Based on an academic suggested textbook	12%
Mandated by college or department	4%
Use my own materials	3%
None of the above	2%
Authored the book	2%

Most often chosen combination:

Based on industry experience, feedback, or trends and on what I feel the course should cover

Question 4: How is your Systems Analysis & Design Course delivered?

Table 4 How SA&D Course is Delivered

Traditional classroom	78%
Hybrid (part classroom/part online)	11%
Online	7%
Some sections online and some in traditional classroom	2%
Other (learner centered approach, videos, etc)	2%

Question 5: What phases of the systems development life cycle are covered in your Systems Analysis & Design course?

(Multiple Answers Allowed)

Table 5 Phases Covered in SA&D Course

Initiation	85%
Planning	92%
Analysis	98%
Design	93%
Implementation	75%
Maintenance	52%
None of the above	0%
Other (testing, project management, non-traditional)	10%

Most often chosen combination:

Initiation, Planning, Analysis, Design, Implementation, and Maintenance

Question 6: What system development approaches do you cover in your Systems Analysis & Design Course?

Table 6 Approaches Covered in SA&D Course

Both traditional and object oriented	53%
Traditional	25%
Object Oriented	15%
Traditional, object oriented, and other (Agile, RAD, JAD, etc.)	5%
Other (Method Engineering, Short life cycle, prototyping)	2%

Question 7: What system development methodologies or models do you cover in your Systems Analysis & Design Course? (Multiple Answers Allowed)

Table 7 Methodologies Covered in SA&D Course

Waterfall	80%
Boehm’s Spiral	24%

Prototyping	81%
Object Oriented	66%
Rapid Application Development	75%
Extreme Programming	35%
Scrum	12%
None of the above	0%
Other	10%

Most often chosen combination:

Waterfall, Prototyping, Object Oriented, and Rapid Application Development

Question 8: What project feasibility measurement concepts and techniques do you cover in your Systems Analysis & Design Course? (Multiple Answers Allowed)

Table 8 Feasibility Concepts Covered

Economic Feasibility	84%
Technical Feasibility	89%
Organizational/Cultural Feasibility	71%
Resource Feasibility	56%
Scheduling Feasibility	63%
Cost/Benefit Analysis	77%
Return on Investment	62%
None of the above	7%
Other	5%

Most often chosen combination:

Economic Feasibility, Technical Feasibility, Organizational/Cultural Feasibility, Resource Feasibility, Scheduling Feasibility, Cost/Benefit Analysis, Return on Investment

Question 9: What project management tools/techniques do you cover in your Systems Analysis & Design Course? (Multiple Answers Allowed)

Table 9 Project Management Tools/Techniques Covered

Microsoft Project	44%
Work Breakdown Structures	30%
GANTT Charts	66%
PERT Charts	56%
Critical Path	51%
None of the above	17%
Other	7%

Most often chosen combination:

Microsoft Project, GANTT Charts, PERT Charts, Critical Path

Question 10: What information gathering techniques do you cover in your Systems Analysis & Design Course? (Multiple Answers Allowed)

Table 10 Information Gathering Techniques Covered

Interviews	94%
Questionnaires	84%
Observation	80%
Heuristic Analysis	17%
Protocol Analysis	17%
Document Review	77%
JAD	58%
None of the above	2%
Other	7%

Most often chosen combination:

Interviews, Questionnaires, Observation, Document Review, JAD

Question 11: What diagramming techniques do you cover in your Systems Analysis & Design Course? (Multiple Answers Allowed)

`**Table 11** Diagramming Techniques Covered in SA&D Course

E-R Diagrams	77%
Data Flow Diagrams	81%
Flowcharts	30%
Structure Charts	39%
Database Diagrams	34%
UML Class Diagrams	52%
UML Use Case Diagrams	54%
UML Activity Diagrams	34%
UML Communication/Collaboration Diagrams	23%
UML State Machine Diagrams	22%

Package Diagrams	9%
None of the above	1%
Other	6%

Most often chosen combination:

E-R Diagrams, Data Flow Diagrams

Question 12: What other system development concepts and techniques do you cover in your Systems Analysis & Design Course?

(Multiple Answers Allowed)

Table 12 Other System Development Concepts and Techniques Covered

Systems Development Life Cycle	91%
Interface Design	68%
Forms Design	55%
Database Design	58%
Network Design	21%
Buy vs. Build	63%
Object and Class Design	42%
Use Case Descriptions	62%
UML	39%
Modular Concepts (cohesion and coupling)	33%
People and Resistance Issues	54%
Scope Creep	58%
Pseudo code Techniques	20%
Structured English	28%
None of the above	1%
Other	7%

Most often chosen combination:

Systems Development Life Cycle, Interface Design, Forms Design, Database Design, Network Design, Buy vs. Build, Object and Class Design, Use Case Descriptions, UML Modular Concepts (cohesion and coupling), People and Resistance Issues, Scope Creep, Pseudo code Techniques

These next tables represent the answers given to a series of follow up questions that were administered to the survey respondents.

Question 13: Is your course delivered in one course or two**?**

Table 13 Number of Courses

One	76%
Two	24%

Question 14: Do you have a course project?

Table 14 Respondents With Course Project

Yes	96%
No	4%

Question 15: Do you use a real world or simulated project?

Table 15 Real Or Simulated Project

Real	58%
Simulated	42%

Question 16: If real world, how do you find the projects?

Table 16 How Projects Are Found

Instructor finds projects	40%
Students find projects	60%

Question 17: Do you split students into groups or do all students work on one project

Table 17 How Students Collaborate On Project?

Split into groups	88%
All work together	7%
Students work individually	5%

Question 18: Does the course project extend beyond the course and one semester

Table 18 Does Course Extend Beyond One Semester?

Yes	21%
No	79%

Question 19: In your SA&D course, do you use more lecture or hands-on activities?

Table 19 Lectures Or Hands On

Lecture	22%
Hands on	9%
About Equal	69%

Question 20: Do you feel that the purpose of a SA&D course should be to give students practical experience or theoretical foundation?

Table 20 Instructors Perception of the Purpose of the SA&D Course

Practical experience	4%
Theoretical Foundation	2%
Mostly Practical	31%
Mostly Theory	22%
Even Split	40%

5. CONCLUSION

The results of this research seem to indicate that, contrary to the opinion of Harris et al., for the most part there is a consensus on how best to teach the undergraduate systems analysis and design course. There are overwhelming majority answers to almost all of the survey questions.

However, there are some results that warrant additional discussion. First is the question as to what system development approaches are covered in the course. A majority of the respondents (58% total) cover both the traditional and object-oriented approaches. This lends credibility to the instructors' faith in the importance of teaching both approaches. Surprisingly, 25% of the respondents teach only the traditional approach and 15% teach only the object-oriented approach in their course. This is surprising given the popularity of both the traditional and object-oriented approaches in industry (Satzinger, Jackson, and Burd 2009).

Question two, which asks what textbook the instructors' are using in their SA&D course shows an area of heterogeneity. While Whitten & Bentley, "Systems Analysis and Design Methods" is the most often used book by respondents in this survey, it is clear that there are many popular books. It is interesting that five percent of the respondents chose to use no book and just use their own materials.

Another question that warrants some discussion is question four, which asks how the course is delivered. Although the online format has gained some traction, the overwhelming majority (78%) still deliver their SA&D course in a traditional classroom setting. So, depending on your viewpoint, that 78% number may be surprisingly high, surprisingly low, or just about right.

It appears that three quarters of the respondents deliver all of their course material in one class (although many of the respondents commented that they wished they had two). Almost all of the respondents (96%) have a course project although they were split as to whether to use a real world project or a simulated project (such as a case study).

When asked if they used more lectures or more hands on activities in their courses, the vast majority of respondents (69%) said that they used both about evenly. Given the statistics on the commonality of lecturing (Griffiths and Oates 2003), it was a bit surprising that only 22% said that they more often utilized a lecture oriented approach.

Perhaps the one item that demonstrates that there is some fragmentation of the instructors' beliefs about the systems analysis and design course is the last question that asks if the purpose of a SA&D course should be to give students practical experience or theoretical foundation. Although a slight majority felt that the focus of the course should be evenly split between both, there were a large number of respondents who felt that the course should be exclusively oriented towards either practical experience or theoretical foundation.

This paper was intended to be an introductory seminal work. Future research will focus on two questions. First, does the demographic background of the instructor (in terms of industry experience, years teaching, etc.) affect the course content of the SA&D course? Second, and perhaps more importantly, is the content of the typical undergraduate SA&D course consistent with the skills, tools, and knowledge required in industry?

6. REFERENCES

Avison, D. and Cole, M. (2006), "Reflections on Teaching Information Systems Analysis and Design: From Then to Now!," *Journal of Information Systems Education* (17:3), pp. 253--256.

1. *Burns*, T., *Klashner*, R. (*2005*), "A Cross-Collegiate Analysis of Software Development Course Content", Proceedings of the 6th Conference on Information Technology

Education, Newark, NJ, USA, pp**. 333-**337. 2005.

Glaser, B. G., & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research.* New York: Aldine Publishing Company.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D., & Longenecker, H., (2002), "IS 2002 : Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems," Communications of the AIS, Vol11, No 1.

Griffiths, G., Oates, B., (2003) "Lecture-free Teaching or Systems Analysis: An Action Research Study", *Informing Science*

Harris, A., Lang, M., Oates, B., & Siau, K. (2006) "Systems Analysis and Design: An Essential Part of IS Education". *Journal of Information Systems Education*, 17(4): 241-248. (Special Issue on "Systems Analysis and Design Education")

Rob, Mohammad A. (2006), "Dilemma between The Structured and Object-Oriented Approaches to Systems Analysis and Design", *Journal of Computer Information Systems*. Vol. 46, Issue 3, Spring 2006, pp. 32-41.

van Vliet, P. J. A. and Pietron L. R. (2006), "Information Systems Development Education in the Real World - A Project Methodology and Assessment", *Journal of Information Systems Education*, 17, 3, 285--293.

Appendix I

Respondents' answers to the question: "What other concepts, methods, models, approaches, tools, and techniques do you cover that have not been mentioned in this survey?"

Accounting for advanced analytic applications and their impact on data warehousing & Mart design; dimensional modeling and conversion of ERD to dimensional models
ARIS enterprise model
Case based approach
Case tool: visible analyst
Critical requirements analysis, requirements traceability, presentation skills, documentation
Crud matrix, application architecture design, strategic information planning, Zachman framework
Data management (meta data, data analysis, data governance and stewardship)
Data normalization, unit testing, system testing, usability testing, acceptance testing
Dialog diagrams
Each project is different and one must design an approach to use to complete a project
Error, Fault, Failure and how to identify and classify IS problems. Report Design, Structured Walkthroughs
Ethical analysis
Higher abstraction of SAD: meta-modeling and method engineering
I focus on Requirements analysis using a use case approach. I cover CRC cards as a method to design system architecture - what classes are needed and what are their responsibilities.
Logical process modeling; logic modeling
Management issues
Management of Change, ANT, Socio-technical, Design of experiments
MS Visio
Naive inquiry, action methods
Participative approach and even EUC
Physical architecture design, systems controls
Process modeling
Project monitoring using earned value measurement, service oriented architecture (but these are covered in different modules of our programme)
Reducing risk by downloading, testing, and perhaps modifying open source software.
RUP process, Analysis patterns, Design Patterns (Larman, GOF, Buchmann), Arquitectural issues (horizontal/vertical partitioning, architectural patterns)
Selling projects, tradeoff triangle, team roles and conflicts
Soft skills - written and oral communication, team skills, presentation skills
Soft systems methodology (checkland)
Systems analysis is a people sport--we spend time on Maxwell's leadership, on the works of Stephen Covey, on team development skills, meetings, and on organizational development and reengineering concepts.
Systems thinking, Added value
Team-based, real-world projects with local clients
Teamwork and experiential learning, group issues, experts not wanting to make theselves obsolete
Testing, support and maintenance, usability, user centered design techniques, navigation diagrams, training, change management
The real world, Office Space
The students are required to develop a sample system analysis & design document based on instructor developed material
Vendor scorecards, estimating techniques, creating interview outlines
Writing requirements, Planning Analysis, Analyst skills and role, Stakeholder analysis, Enterprise Analysis, Estimation, CHAOS study, Quality Assurance

Appendix II

Respondents' answers to the question: "How do you balance the challenge of delivering the volume of material required for a typical SA&D course and keeping students from being overwhelmed/bored?"

It is a challenge since the concepts are abstract and student experience is minimal.
Try to make them work in teams, discuss among them, talk to people about real projects
Make the project the focus of the course, eliminate duplicate assignments, not tests
Best balance is to make sure they get some time in the lab to work on their projects; the hands-on components.
Make class interesting; give exams that require demonstration of learning how to apply concepts.
2 semesters help
They really get into their projects
After 25 years I think I balance it well. Hands on activities prevent boredom.
Keeping the students engaged / involved with discussion and hands on activities helps.
I decided to go with a textbook that covered the SDLC but with only 10 chapters.
Use of animation in presenting modeling, use of real world example, guest speakers, and hands on practice
By making them read at home and applying the knowledge in class students learn fast and efficient.
Selecting only relevant chapters from textbook, not discussing others.
Mixture of course delivery methods – exercises, practical examples, class participation etc.
This is a big challenge. I try to use just-in time learning, so they learn how to do something in class and then apply that technique for their client. The project is divided into 6 milestones: team establishment, problem id, current system processes, proposed system, final report, final presentation/poster. This is crucial.
I try to be judicious about what I cover. Based on my discussions with colleagues/adjuncts, I try to cover the material that is currently important. Over the course of two semesters, I think I get most of the important stuff. Are they overwhelmed? Maybe. Are they bored? Well, life isn't always fun, is it?
In the first course I concentrate totally on the Planning and Analysis phases. I emphasize the importance of project management (work plan, cost benefit analysis, breakeven) and on business modeling using UML. I focus on the Use Case, Class and Activity diagrams. I do not pursue design topics until the second course.
By keeping them involved and sometimes be relaxed on some issues like deadlines.
What works best is to scale the project down to a manageable level, this also allows for a discussion on scope and leads to the class developing a statement of scope
I give up some course content via lectures to focus on the project.
I view the course as a design course.
This is the reason for split into 2 courses
I have taught the course for 12 years and it has been a learning experience. I develop a schedule that is planned out for each class period. I don't try to cover the entire text book. I pick topics that I feel will help them the most in the real world. There is a link to the course outline above.
Keep a good balance of conceptual and practical/hands-on
The hands on activities focus most students.
Cut down on the volume. Cannot cover everything and do it well. Better to do less

Explain the material and then let them do a small in-class assignment in a group
30 years of program manager experience
I have found that carefully selected smaller chunks gone over in depth is far more effective than a voluminous broad overview.
The lectures are very short, maybe 5-10 mins followed by discussion or hands-on.
By engaging them in discussion and activities
We focus on the deliverables each week

Appendix III

Respondents' answers when prompted to: "List some hands-on activities you do in your SA&D Course"

Gathering requirements, designing a solution, coding, testing
Research project: best analysis and design practices in real companies: which problems they find while developing software, how do they solve/try to prevent them, etc.
Lots of exercises on design patterns. Working in groups, they present the solutions to a selection of problems.
Each of the deliverables in the SDLC is explored with the final outcome a deliverable; might use project management software, CASE tool, etc.
Perform and document varíous analysis and design tasks. Program, test and document (parts) of the system
They use Microsoft Visio to draw the dfds; VB for the screenshots; Word for the narratives; and Excel for the reports.
Draw diagrams on the board in class; have students evaluate each others' work; practice modeling and defending answers for choices include in models; in class discussions to provide support for why one way of modeling a particular problem provides a better or more useful approach than another; CRC cards; act out CRC exercise;
Practice with case tool- Visible Analyst, make decision table, calculate cash flow, find solutions to cases
Dfd, erm
I have a hands on activity for most of the important topics. Sometimes I use an activity from the back of the book, other times I create something.
Uml modeling
DFD, interviewing, reviewing each other project ideas (small groups)
In class exercises on Process and data modeling, in class Lab time to work on projects
Work on project during class meetings
Theory, exercises or case studies on different types of models, cost/benefit exercises etc.
Modeling problems, interview role playing, mini-cases
I have them work on the milestones for their projects (data flow diagrams, data models, requirements analysis plans, etc.
Comprehensive group project throughout the semester going through the all the activities of SDLC, three-to-four presentations on key areas of learning/SDLC (problem definition, process modeling, data modeling and program design), use of tools such as Microsoft Project, Visio, Visible Analyst, as well as develop all documents necessary to go through the SDLC activities. Also write two research papers, and sometimes in-class case studies from the book. Most emphasis is on data-flow diagrams that students only learn from this class.
Activity Diagrams, Data Flow Diagrams, ERD's, interface prototypes and implementation recommendations.
In addition they interview user(s) from the project and conduct a needs analysis, they also present their projects to the department faculty
Project, interviews, data collection, analysis, data flow diagrams, project design, meetings with clients
The students learn to use CASE tools.
SAD modeling methods
Modeling problems in class: dfd, erd, data dictionary. Process specs, use-case diagramming.
The in-class project described above is the main in-class hands on.
Dfds, interviewing

Give class time for students to work on their projects and work on the interfaces between their various projects, also have several in-class worksheets for dfds
Games, mini cases, exercises
Practicing modeling techniques, doing presentations and walk-throughs
Create project documents
Erds, dfds, Data models
Assignments on Visible Analyst and Access, plus they work on their projects
Data flow, project mgt
Requirements classification "game"
Design the worst interface you can
Mock client requirements interview
Break into a website (application security)
We do some sample parts of the projects on a simplified case study so they are aware of pitfalls and issues that may arise.
Project request, project plan, use case model, class diagram, sequence diagrams, object state charts, normalized data model
Data-flow diagramming, project management software, report development
Create a data model, create a process model,   Read current articles that pertain to the topic of the week, and disucss them in class.
Class discussions and exercises, homework assignments on dfds, ERD, problem definition, proposed design

Appendix IV

Respondents' answers to the question: "Do you have any other techniques, comments, or concerns that you would like to share with this study?"

The flow of the book should be such that a student can logically progress from business case to a simple design document. My suggestion will be to deemphasize database design, interface design etc since there is no time for these topics. Database design should be done in a database course anyway. Include risk management
Since the course is so heavy on team work ensure that there is a evaluation rubric for team participation; everyone evaluates the time/effort put forth by each member of their team, will eliminate the social loafer from earning the same grade as the rest of the team.
I think that a 2-semester sequence would be very helpful
I would like to see Agile (scrum) featured in a textbook. Even though some businesses would not go with this methodology I think in a web class my students could get through some aspect of a project using Agile in a single semester. With students living in diverse locations a collaborative environment could work with a Agile project.
Business rules are also extremely important in our class,
Blend of standard textbooks with modern examples (facebook, Nintendo Wii, Twitter, iphone etc) works very well because of the deep familiarity of the students.
Use many examples from industry in class discussions.
Interacting with a real client is invaluable.
I think the key is to make the material as practical as possible. I come from industry, and I have lots of war stories, which seem to help. I also bring in guest speakers from industry to back up and extend what I talk about. I get good feedback from those classes.
I have my student-groups develop almost every document that I discuss in the class and submit at the end of the semester as a portfolio.
The course projects are completed in groups with every attempt made to get a cross section of majors in each group
Our Advisory Board saw a shortfall in our previous program – their observation was that students need to work in a large systems environment during some part of their coursework – not just small-scale stand alone projects.
We are fortunate to approach the course sequence as team of faculty rather than using a silo-based approach to the courses in the MIS major.
At another school at which I taught this was spread over 2 semesters and it worked much better.
This is our capstone sequence and fulfills many of our ABET outcomes (we are ABET accredited) so it's a very important sequence and we put our best teachers into it
The only way students learn the techniques is to do them
If the course is in a business school for students who seek to become managers, being able to recognize data and how it traverses the company is more important than creating a design that is ready for the programmers. Hence, the use of object modeling may not be the most effective. Data flow diagramming is one technique that is both useful and can be easily understood by others in presentations. If the sa&d course is directed to cs students, then object notation might be best.
I mix theory and hands-on in every class to make it interesting. Only theory makes it boring.
The main problem I encounter is convincing the students of the relevance and importance of the material -- often they seem disinterested
While doing a real project allows the students to see the difficulty of getting requirements with clients, it is difficult to control.

A Relational Algebra Query Language For Programming Relational Databases

Kirby McMaster
kmcmaster@weber.edu
CS Dept., Weber State University
Ogden, Utah 84408 USA

Samuel Sambasivam
ssambasivam@apu.edu
CS Dept., Azusa Pacific University
Azusa, California 91702 USA

Nicole Anderson
nanderson@winona.edu
CS Dept., Winona State University
Winona, Minnesota 55987 USA

Abstract

In this paper, we describe a Relational Algebra Query Language (RAQL) and Relational Algebra Query (RAQ) software product we have developed that allows database instructors to teach relational algebra through programming. Instead of defining query operations using mathematical notation (the approach commonly taken in database textbooks), students write RAQL query programs as sequences of relational algebra function calls. The RAQ software allows RAQL programs to be run interactively, so that students can view the results of RA operations. Thus, students can learn relational algebra in a manner similar to learning SQL—by writing code and watching it run.

Keywords: database, query, relational algebra, programming, SQL

1. INTRODUCTION

Most commercial database systems are based on the relational data model. Recent editions of database textbooks focus primarily on the relational model. In this dual context, the relational model for data should be considered the most important concept in an introductory database course.

The heart of the relational model is a set of objects called relations or tables, plus a set of operations on these objects (Codd, 1972). Coverage of the relational model in database courses includes the structure of tables, integrity constraints, links between tables, and data manipulation operations (data entry and queries).

Classroom discussion of queries and query languages generally leads to a detailed presentation of SQL. *Relational algebra* (RA) as a query language receives less attention. In a survey of database educators, Robbert and

Ricardo (2003) found that only 70% included RA in their courses, compared to 92% for SQL.

Database textbooks provide substantially more material on SQL than on RA. An extreme case is the textbook by Hoffer, et al (2008), which provides two full chapters on SQL but does not mention RA.

Why Teach Relational Algebra?

There is almost universal agreement that SQL is an essential component of an introductory database course. But should we also teach relational algebra? There are several good reasons for doing so.

1. The main reason for teaching RA is to help students better understand the relational model. At the conceptual level, the relational model provides a flexible, adaptable way to query a database. The organization of data into tables, together with RA operations, provides the foundation for this flexibility.

Relational algebra is a *query* language, not a database design tool. However, an understanding of how RA operations can be performed on tables to extract information should help support database analysis and design decisions.

2. Knowledge of RA facilitates teaching and learning SQL as a query language. The basic syntax of the SQL SELECT statement provides an integrated framework for combining RA operations to express a query.

3. An understanding of RA can also be used to improve query performance. The query-processing component of a DBMS engine translates SQL code into a query plan that includes RA operations. The DBMS query optimizer, together with the database administer, can speed up query execution by reducing the processing time of the RA operations.

How to Teach Relational Algebra?

If an instructor decides to include relational algebra as a topic in a database course, a follow-up question is how to present this topic to students? RA coverage in leading database textbooks often takes a *mathematical* approach. For example, the texts by Connolly and Begg (2009), Elmasri and Navathe (2006), and Silberschatz, et al (2006) present RA concepts using mathematical notation. There are several problems with this form of representation.

Many database students are not comfortable with mathematical notation, such as the use of Greek letters (e.g. σ and π) in a new context. The mathematical approach often mixes *infix* notation (operator name is placed *between* two operands; e.g. table1 union table2) and *functional* notation (operator name is placed *before* the operands; e.g. project table3 cols) when performing multiple RA operations within a single expression. This makes the expressions difficult to interpret, and it disguises the procedural nature of RA.

More importantly, students cannot execute query programs written in the mathematical notation. There is no easy way to verify that the mathematical description of a query is correct.

The mathematical approach contrasts with how *programming* courses are taught. In a programming course, an important part of learning occurs when students write instructions for the computer and watch their code run. Errors in program execution provide feedback, which reduces the gap between a student's perception of the problem and how the computer interprets the proposed solution.

To demonstrate how computer implementations differ from mathematical models, students need software to experiment with. Students learn mathematical and computational concepts more effectively when they can work with actual computer representations. As with other programming languages, this principle applies when we teach students how to query using relational algebra.

All major relational database products offer SQL as the primary query language. On the other hand, very few computer environments are available for developing and running RA programs. One database system to offer RA as a query language is LEAP (Leyton, 2010). The Rel DBMS (Voorhis, 2010) uses a form of RA called Tutorial D (Date and Darwen, 2007). A third choice is WinRDBI (Dietrich, 2001), which supports queries using RA and other query languages.

Each of the above systems enables RA queries within a specific database system. None allow you to use RA to query desktop databases. In this paper, we introduce a Relational Algebra Query Language (RAQL) and a custom Relational Algebra Query (RAQ) software product that can be used to query relational databases.

We first present a function-based syntax for writing RAQL query programs as sequences of RA operations. We outline the main features of the RAQ software. Next, we demonstrate how to use the software to execute RAQL programs. Finally, we give examples of RA concepts that can be taught using this approach.

2. RELATIONAL ALGEBRA PROGRAMMING

A RAQL query program consists of a set of statements that specify operations to perform on database tables. The statements are executed in a particular sequence to yield a result table that satisfies a query. Each statement might consist of a single relational algebra operation, or several operations can be combined into one "algebraic" expression. Rather than use complex expressions in query programs, we prefer to have each line of code perform a single RA operation. Our coding style reflects 2GL (assembly language) thinking more than 3GL thinking (e.g. Fortran, C).

We provide a library function for each RA operation. A RAQL program is written as a sequence of RA function calls. Each function has one or two input parameters that are tables, plus other input parameters as necessary. The output of each function is another table. Using functions to implement RA operations provides database students with a comfortable programming environment for creating RAQL query programs.

Functions are provided for the following relational algebra operations:

Table 1: Relational Algebra Functions

Operation	Function
selection	TSelect(Table1,RowCondition)
projection	TProject(Table1,ColumnList)
join	TJoin(Table1,Table2,JoinCondition)
union	TUnion(Table1,Table2)
intersection	TIntersect(Table1,Table2)
difference	TMinus(Table1,Table2)
product	TProduct(Table1,Table2)
division	TDivide(Table1,Table2)
rename	TRename(Table1,OldColumnName, NewColumnName)

To illustrate programming using RA functions, we require a sample database. The structure of a simple inventory database is described in the next section.

Relational Database Example

Suppose an INVENTORY database for a manufacturing environment consists of two tables, STOCK and STKTYPE. The diagram in Figure 1 describes the relational model for this database.

This data model assumes that inventory items are divided into categories, or types. Attributes that apply to individual items are recorded in the STOCK table. Attributes that apply to all items of the same type are included in the STKTYPE table. The two tables are linked by a common type code (SType and TType).

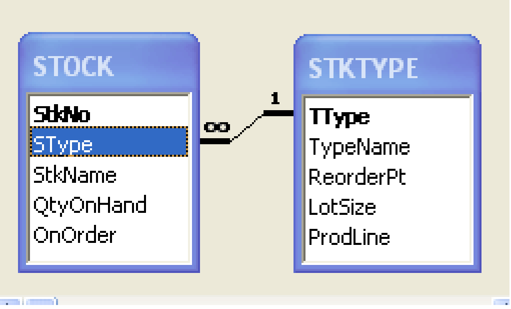


Figure 1: Inventory Database
Primary keys are shown in **bold**

In this basic system, when the quantity-on-hand for an item drops below its reorder point, a production run of a predetermined lot-size is scheduled on a specific production line. It is assumed that reorder point, lot-size, and production line depend on the stock type rather than on the individual item. Whenever a production run is scheduled, the OnOrder field for the item is set to 'Y'. This field is reset to 'N' after the order is filled.

RA Query1 Program

Consider the following query for the INVENTORY database.

Query1: List the stock number, name, and quantity-on-hand for all items that are manufactured on production line 3.

A RAQL program for this query takes the form of a sequence of Table 1 function calls. Each function receives one or two tables as arguments and returns a temporary table. The temporary table can be used in later RA operations. Sample code for this query is shown below:

```
-- RA Query1: Inventory Query - Line 3
T1 = TJoin('STOCK', 'STKTYPE', "SType=TType")
T2 = TSelect(T1, "ProdLine=3")
T3 = TProject(T2, "StkNo,StkName,QtyOnHand")
```

An explanation of each line of code for this program follows:

Line 1: This is a comment (--)

Line 2: The STOCK and STKTYPE tables are *joined*. The join condition states that the SType field in the STOCK table must match the TType field in the STKTYPE table. Actual table names are placed in matching single (or double) quotes, since they are fixed string values. The join condition is also placed in quotes. The output of the TJoin function is a *cursor* (a temporary table in memory). The cursor name is randomly generated and is assigned to variable T1. The name of the cursor is unknown to the programmer, but the cursor can be referred to in later program statements using the variable name.

Line 3: Rows of cursor T1 are then *selected* if they satisfy the condition that the ProdLine (production line) field in T1 equals 3. Quotes are not needed for the number 3. If quotes are needed inside a row condition, then single and double quotes should be nested in pairs (e.g. "OnOrder='N'"). The output cursor name is assigned to variable T2. The T1 argument is not placed in quotes, since T1 is a variable.

Line 4: The three columns of cursor T2 specified in the column list are *projected* as cursor T3, which is the final result table for the query.

3. RAQ COMPUTER SOFTWARE

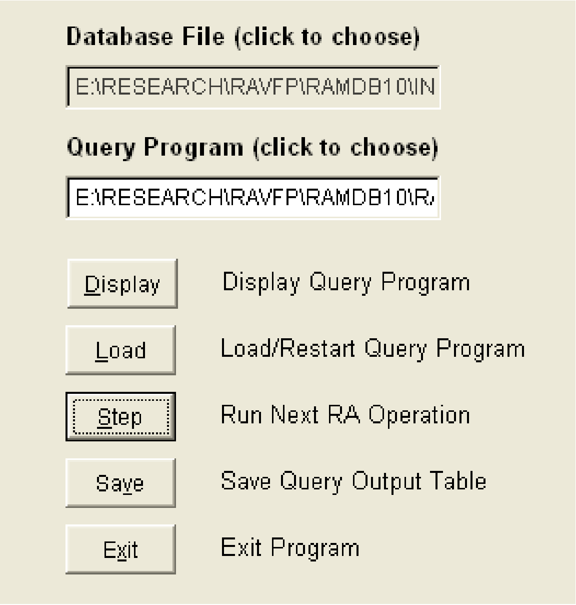
The RAQ software allows us to execute queries written in the format of Query1. Our explanation of how to use RAQ to perform queries is organized according to the controls (textboxes and command buttons) on the RAQ main screen (see Figure 2).

1. *Database File* textbox: Choose a database file. The database must be in a Microsoft Access MDB file. The file can be selected with the file-chooser dialog box, which includes the ability to search in subdirectories. No other actions can be performed in the RAQ software until a valid database file is opened. Once a database is open, it cannot be changed without exiting and then rerunning the RAQ software.

2. *Query Program* textbox: Choose a RAQL query program. The program must be in a text file having a TXT extension. A new query program can be selected at any time during the execution of the RAQ software, but the actions that follow must be repeated.

3. *Display* button: Display the RAQL program code in a window. This command can be selected whenever the Display button is enabled. Press the Escape key to close the window. The display window is read-only. Any changes or corrections to the RAQL program must be made in a separate text editor.

4. *Load* button: Before a RAQL program can be run, it must be loaded. This action can be repeated when you want to restart the program from the beginning.



Database File (click to choose)
E:\RESEARCH\RAVFP\RAMDB10\IN

Query Program (click to choose)
E:\RESEARCH\RAVFP\RAMDB10\R/

Figure 2: RAQ Program Main Screen

5. *Step* button: Each click of this button executes one RAQL instruction. This will normally be a single relational algebra operation. Comments in the program code are skipped. For each successful RA operation, the resulting query output cursor is shown on the screen. Hit the escape key to close this view.

If an error occurs while trying to execute an instruction, an error message is displayed in the top right-hand corner of the screen. The error message shows the code for the line that was just attempted. Clicking the Display button allows the user to see the error in the context of the full RAQL program.

6. *Save* button: When an operation has completed, the current output cursor can be saved to disk. The format of the saved file is an Excel XLS file. The name of the output file is the name of the RAQL program file, followed by the step number. The contents of the XLS file can be easily transferred to a word processing document or other data file.

7. *Exit* button: Click this button to exit the program. You will be prompted to confirm this request before the program ends.

The availability of most RAQ menu choices depends on which actions have already occurred during program execution. Textboxes and command buttons are disabled when their selection would be inappropriate. For example, a result table cannot be saved if the current command fails to execute correctly.

Running the RA Query1 Program

The previous discussion of RAQ controls and features was fairly general. To provide a more concrete demonstration, we list below one possible sequence of RAQ actions that could be taken to execute the Query1 program.

1. Load the INVENTORY database file. Sample data for this database is listed in Appendix A.

2. Load the text file that contains Query1. Assume this file is named RA*Query1.txt*.

3. Click the Display button to view the query program code (optional).

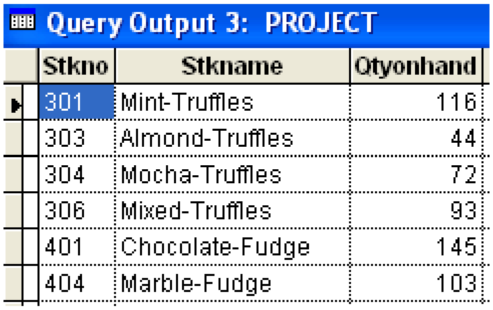
4. Click the Load button to initialize the program.

5. Click the Step button. The comment line will be skipped, and the TJoin operation will be performed. The joined cursor T1 will appear in a window.

6. Click the Step button again, and the TSelect operation will produce cursor T2.

7. Click Step a third time, and TProject will produce and display cursor T3. The final result for Query1 is shown in Figure 3.

8. To save a result cursor, click the Save button after closing the window showing the cursor. If the final result in this example is saved, the output file name will be RA*Query1-3.xls*.



Stkno	Stkname	Qtyonhand
301	Mint-Truffles	116
303	Almond-Truffles	44
304	Mocha-Truffles	72
306	Mixed-Truffles	93
401	Chocolate-Fudge	145
404	Marble-Fudge	103

Figure 3: RA Query1 Final Result

9. Click Exit when you are finished; then confirm when prompted. If you prefer, you can Load and rerun the same RAQL program, or choose a new query program.

Appendix B contains a sample Relational Algebra Project to give students experience writing and running RAQL programs.

4. USING RAQ TO TEACH RELATIONAL ALGEBRA CONCEPTS

The RAQ software can be used to teach relational algebra concepts interactively that are usually explained intuitively. The advantage of using RAQ is that students can visualize the concepts when they are implemented as RAQL programs. Some examples of RA concepts that can benefit from this approach are described next.

Select Before Join

When *select* is used before *join*, the size of the joined table will usually be much smaller than if the join operation is performed first. This will reduce the memory resources required for a query and should decrease processing time. The Query2 RAQL program shown below achieves the Query1 result, but starts with a select operation.

```
-- RA Query2: Select before Join
T1 = TSelect('STKTYPE', "ProdLine=3")
T2 = TJoin('STOCK', T1, "SType=TType")
T3 = TProject(T2, "StkNo,StkName,QtyOnHand")
```

This program can be compared to the Query1 program, where the join operation is performed first. The relative size of the two joined cursors (T1 in Query1 and T2 in Query2) highlights the advantage of joining tables "later."

Set Union and Intersection

The *union* of sets A and B consists of all distinct members of A and B. In RA, the union of two tables does not include duplicate rows. This concept can be illustrated with the following Query3 program.

```
-- RA Query3: Union and Intersection
TA = TSelect('STOCK', "QtyOnHand<50")
TB = TSelect('STOCK', "SType='C'")
T1 = TUnion(TA, TB)
T2 = TIntersect(TA, TB)
```

In this program, cursors TA and TB have identical attribute domains (*union-compatible*). The union cursor T1 does not contain duplicates of rows that satisfy both conditions. The intersection cursor T2 identifies the rows that are in both TA and TB.

Set Intersection and Difference

In set theory, it is known that the relationship between *intersection* and *difference* satisfies the equation

$$A \cap B = A - (A - B)$$

The Query4 program listed below verifies this relationship.

```
-- RA Query4: Intersection and Difference
TA = TSelect('STOCK', "OnOrder='N'")
TB = TSelect('STOCK', "SType='T'")
T1 = TMinus(TA, TB)
T2 = TMinus(TA, T1)
T3 = TIntersect(TA, TB)
```

Here, cursor T1 is A - B, T2 is A - (A - B), and T3 is $A \cap B$. Students can observe that T2 and T3 are identical.

Product vs. Divide

The RA *divide* operation is sometimes described as the "inverse" of the *product* operation, in the sense that for tables A and B,

$$(A \times B) \div B = A$$

The following Query5 program illustrates the nature of this relationship.

```
-- RA Query5: Product and Divide
TA = TSelect('STOCK')
TB = TSelect('STKTYPE')
T1 = TProduct(TA, TB)
T2 = TDivide(T1, TB)
```

In this code, T1 is A x B and T2 is (A x B) ÷ B. Students can note that cursor T2 is the same as cursor TA.

5. SPECIAL CONSIDERATIONS

The RAQ software is not a feature-rich, industrial-strength software product. It was designed for academic use to provide a simple way to teach relational algebra concepts through programming. Users of this software should be aware of certain limitations and special considerations.

1. There is a 100-line maximum for RAQL query programs (not including comments). Each instruction must be on a single line.

2. RAQ provides modest error checking. Error messages are displayed in the upper-right corner of the screen. Messages state the type of error or show the offending line of code.

3. RAQ has limited input options for data. The database must be in an Access MDB (not ACCDB) file. If necessary, convert the ACCDB file to MDB format. More generally, if the database is an ODBC data source (e.g. Oracle, SQL Server, MySQL), then the table structures and data can be *imported* into an Access file before using RAQ.

4. RAQL query programs must be in a text file with a TXT extension. Programs have to be created and modified with a separate text editor, since RAQ does not provide editing capabilities.

5. RAQ output for query results are shown on the screen. The display of intermediate cursors cannot be skipped, but RAQL programs are usually short. Query output can be saved in XLS files, and the Windows operating system provides various print-screen options.

6. For convenience in expressing RA queries, duplicate field names should be avoided in databases. If you prefer to have duplicate field names in separate tables (e.g. the same name for primary key and foreign key), use the TRename function in RAQL programs. This is a constraint inherent in relational algebra (Date, 2004) and not just in our RAQ software.

The SQL SELECT statement allows a field to be specified by including the name of the relevant table (e.g. STOCK.SType). The SELECT statement can do this because intermediate cursors generated in the processing of the statement are invisible and are not referenced. We do not have this luxury in RAQL. Each RAQL statement generates a temporary cursor with an unknown name. If a cursor has a duplicate field name, we cannot "hard-code" the unknown cursor name to identify the field.

7. Nesting of RA function calls within a single statement is permitted but not

recommended. Nested function calls defeat the opportunity to see intermediate result cursors while RA operations are performed. Nesting also disguises the procedural nature of relational algebra.

8. The RAQ software has been tested in Windows XP, Windows Vista, and Windows 7. Administrative privileges may be required for Vista or Windows 7, since RAQ writes some temporary files to the disk.

6. SUMMARY AND CONCLUSIONS

In this paper, we presented arguments for including coverage of relational algebra (RA) along with SQL in database courses. We argued that, in teaching relational algebra to database students, a programming approach is preferable to a mathematical approach. Our recommended programming style is to write query programs in a special Relational Algebra Query Language (RAQL). In this language, query programs are expressed as sequences of function calls, where each call performs one RA operation. Following this format, students gain experience using a procedural query language while learning relational algebra.

Writing query programs improves the educational experience for students, but learning is enhanced if students can execute their query programs. We have developed a custom Relational Algebra Query (RAQ) software environment in which RAQL programs can be run.

The RAQ software allows students to see the intermediate results during the sequence of relational algebra operations. With this capability, students can visualize RA concepts and explore performance issues. Thus, they can learn RA in a manner similar to how they learn SQL—by writing code and watching it run. As Knuth might say, students can better understand a problem by teaching a computer how to solve it (Shustek, 2008).

Note: An executable version of the RAQ program, along with runtime files and the database examples in this paper, can be obtained from the lead author.

7. REFERENCES

Codd, E. F. (1972). Relational Completeness of Data Base Sublanguages. In Rustin, Randall (ed.), Data Base Systems, Courant Computer Science Series 6. Prentice Hall.

Connolly, T., & Begg, C. (2009). Database Systems: A Practical Approach to Design, Implementation, and Management (5th ed). Addison-Wesley.

Date, C. J. (2004). An Introduction to Database Systems (8th ed). Addison-Wesley

Date, C. J., & Darwen, H. (2007). Databases, Types, and the Relational Model (3rd ed). Addison-Wesley.

Dietrich, S. (2001). Understanding Relational Database Query Languages. Prentice Hall.

Elmasri, R., & Navathe, S. (2006). Fundamentals of Database Systems (5th ed). Addison-Wesley.

Hoffer, J., Prescott, M., & Topi, H. (2008). Modern Database Management (9th ed). Prentice Hall.

Leyton, R. (2010). LEAP RDBMS: An Educational Relational Database Management System. leap.sourceforge.net.

Robbert, M., & Ricardo, C. (2003). Trends in the Evolution of the Database Curriculum. *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*. Greece.

Shustek, Len (2008). Donald Knuth: A Life's Work Interrupted. Communications of the ACM.

Silberschatz, A., Korth, H., & Sudarshan, S. (2006). Database System Concepts (5th ed). McGraw Hill.

Voorhis, D. (2010). An Implementation of Date and Darwen's Tutorial D Database Language. dbappbuilder.sourceforge.net /Rel.php.

APPENDIX A: INVENTORY Database – Sample Data

STOCK Table

StkNo	SType	StkName	QtyOnHand	OnOrder
101	B	Prune Basket	65	N
105	B	Pear Basket	48	N
107	B	Peach Basket	21	Y
202	W	Deluxe Tower	54	N
204	W	Special Tower	29	N
301	T	Mint Truffles	116	N
303	T	Almond Truffles	44	Y
304	T	Mocha Truffles	72	N
306	T	Mixed Truffles	93	N
401	F	Chocolate Fudge	145	N
404	F	Marble Fudge	103	N
502	C	Berry CheeseCake	73	N
505	C	Apple CheeseCake	46	N
506	C	Lemon CheeseCake	18	Y
508	C	Plain CheeseCake	65	N

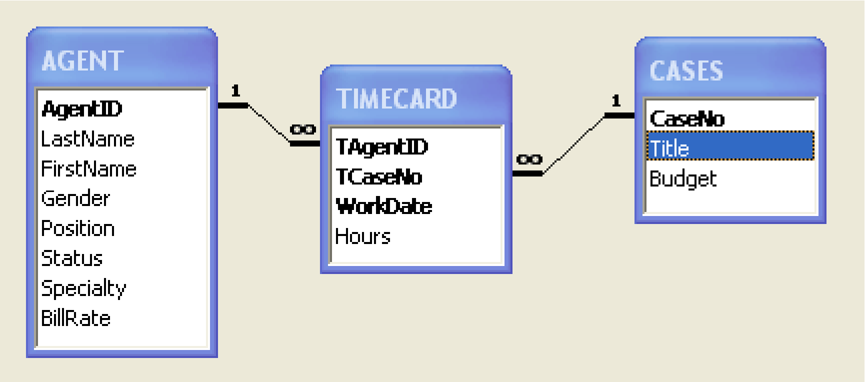
STKTYPE Table

TType	TypeName	ReorderPt	LotSize	ProdLine
B	Basket	60	90	1
C	CheeseCake	50	75	2
F	Fudge	120	180	3
T	Truffles	90	120	3
W	Tower	40	60	1

APPENDIX B: Relational Algebra Project

This project uses a Microsoft Access file that contains a Time-and-Billing database for the XFiles group in the FBI. The file is called XFILES.mdb. The database is used to track the number of hours spent by agents on cases. Each agent fills out her/his time card each day, charging up to 8 hours per day on cases.

The database consists of three tables: AGENT, CASES, and TIMECARD. The relational model for this database is shown in the following diagram.



1. Write a *Relational Algebra Query Language* (RAQL) program for each of the following queries:

   Query 1: List the agent ID, last name, position, and bill rate of all Special Agents that have a bill rate greater than $75 per hour.

   Query 2: List the case number, case title, and budget of all cases that have been worked on by a female agent.

   Query 3: List the agent ID, last name, specialty, and bill rate of all agents that have worked on the Fat-Sucking Vampire case.

   Query 4: List the work date, case title, agent ID, and hours for all time card records where less than 4 hours were charged.

   Query 5: List the last name, first name, and gender of all agents that are female *or* have worked on the Bermuda Triangle case.

   Query 6: List the last name, first name, and gender of all agents that are female *and* have worked on the Bermuda Triangle case.

   Query 7: List the agent ID, last name, and specialty of all agents that have *not* worked on the Dark Matter case.

   Query 8: List the agent ID, last name, and first name of all agents that have worked on every case that Scully has worked on.

2. Use the RAQ software to run your query programs. For each query, save the final output table in an Excel file.

3. Combine your query results in a Word document, grouping together for each query:
   a. The *word definition* of the query.
   b. The *source code* for your RAQL program.
   c. The final *output table* from the query.

The Greening of the Information Systems Curriculum

Patricia Sendall
patricia.sendall@merrimack.edu
Girard School of Business
Merrimack College
N. Andover, MA 01845, USA

Li-Jen Shannon
LYS001@shsu.edu
Computer Science Department
Sam Houston State University
Huntsville, TX 77341, USA

Alan R. Peslak
arp14@psu.edu
Information Sciences & Technology
Penn State University
Dunmore, PA 18512, USA

Bruce Saulnier
bruce.saulnier@quinnipiac.edu
Computer Information Systems
Quinnipiac University
Hamden, CT 06518, USA

Abstract

The purpose of this study is related to Green Information Technology (IT), Green Computing, and/or Sustainability (GITS) curriculum initiatives in institutions of higher education in the US and abroad. The purpose of this study is threefold; 1) to evaluate whether GITS academic programs have been initiated in higher educational organizations; 2) to analyze if GITS programs and/or courses are currently in place; and 3) to investigate the conceptual framework of GITS across campuses.

Keywords: Green IT, Green Computing, Sustainability, Curriculum Design, Higher Education

1. INTRODUCTION

According to Wheeland (2009), in the world of information technology (IT), there is a likelihood of advancement for employees who are well versed in “green” issues. Lyon, Dorney & Sargent (2010) define the green economy as:

> *…economic activity generated by companies, customers and the public sector in the form of products, services, and business models that promote economic growth, reduce*

environmental impacts and improve social well being (pg. 1).

The *State of Green Business 2009 Report notes that many college graduates and experienced professionals are looking to join organizations that combine the entrepreneurial vitality of a for-profit enterprise with the benevolence and impact of a non-profit. These green socially responsible enterprises should prosper and help develop a distinctive, emerging green workforce* (Lyon, et al, 2010).

Environmental responsibility is becoming a necessary topic for information technology (IT) enterprises and their suppliers of technology. According to Mines & Davis (2007), corporate IT will go green in the coming years, "driven by a combination of cost efficiency, regulatory compliance, and corporate responsibility motivations." (pg. 1).

It is clear that corporations, governments and educators are taking note of the impact of their own resource consumption on the environment. For example:

- Twenty-five percent of the world's population is consuming 70-80% of the world's resources (ACUPCC, 2009).
- By 2050 we will have 9 billion people and economic output that is increased 4-5 times (ACUPCC, 2009).
- According to McKinsey & Company, because of the increasing need for computational power, data storage and communications, the technology footprint amounts to about 2% of global greenhouse gas emissions, and is estimated to increase to 3% by 2020 (Makower, 2010).
- The U.S. has approximately 5% of the world's population and is consuming 25% of the world's resources (Education for Sustainable Development, n.d.).

The report on *The State of Green Business 2010* revealed positive indicators for the green economy, including environmental, health and safety (EHS) spending, employment, and green product development. According to Makower (2010), eighty-three percent said their 2010 EHS spending will be equal to or greater than 2009, with nearly half (43 percent) reporting an increase in spending. The good news is that hiring freezes are thawing. Open requisitions for environmental- and sustainability-related positions rose from 18% in mid-2009 to 23% in early 2010. Finally, investments in green product development continue to show growth, especially in companies with revenues greater than US $1 billion.

Not only did sustainability practices not go away amid the harsh economic environment, they actually survived and thrived, according to the report on *The State of Green Business 2010* (Makower, 2010). In addition, approximately 86 percent of U.S. companies said they planned to continue to invest in green products and programs in equal amounts as they did in the year prior (Makower, 2009).

Sustainability and Green IT

Not everyone agrees on what defines sustainability. Arizona State University's Global Institute of Sustainability defines a sustainable society as one that,

…considers the interconnectedness of environmental, economic, and social systems; reconciles the planet's environmental needs with development needs over the long term; and avoids irreversible commitments that constrain future generations. (*What is Sustainability,* para 1).

Lyon, et al (2010) defines sustainability after Rosenbaum (1993) and Vieira (1993) as,
…using methods, systems and materials that won't deplete resources or harm natural cycles and as a concept and attitude in development that looks at natural land, water, and energy resources as integral aspects of development; sustainability also integrates natural systems with human patterns and celebrates continuity, uniqueness and stewardship (pg. 3).

In its most general sense, sustainability refers to the *capacity to endure*. In ecological parlance, the word describes how biological systems maintain their well being over time. More specifically, for humans it refers to the potential for long-term maintenance of well-being, which is dependent upon the well-being of the natural world and the responsible use of natural resources.

There is abundant evidence that humanity is living unsustainably. As the earth's human population has increased natural ecosystems have declined and changes in the balance of natural cycles have had a negative impact on both humans and other living systems. Returning human use of natural resources to within sustainable limits will require a major collective effort on the part of all humanity (Adams & Jeanrenaud, 2008).

There are ways of living more sustainably and they can take many forms, such as: reorganizing living conditions (eco-villages, sustainable cities, etc); reappraising economic sectors (green building, sustainable agriculture, etc.) or work practices (sustainable architecture); using science to develop new technologies (green technologies, renewable energy, etc.); and adjusting individual lifestyles to conserve natural resources.

Green IT or green computing is a subset of sustainability. According to Lamb (2009), *"Green IT is the study and practice of using computing resources efficiently. Typically, technological systems or computing products that incorporate green computing principles take into account the so-called triple bottom line of economic viability, social responsibility, and environmental impact. This approach differs somewhat from traditional or standard business practices that focus mainly on the economic viability or economic benefits rendered by a computing solution."* (pg. xxiv).

According to Ruth (2009), the "green computing" idea started in 1992 when the US Environmental Protection Agency (EPA) launched Energy Star. The term *green computing* has evolved over the past decade (Choi & Mata-Toledo, 2009). While there is no universal agreement on the definition of *green computing*, the authors consider it to encompass all aspects of computer technology that contribute to the reduction of global warming and e-waste through "alternative energy sources, power management, recycling and biodegradable materials." (pg. 1).

Andy Hopper, a professor of computer technology at the University of Cambridge defines the green computing movement as, "a multi-faceted, global effort to reduce energy consumption and promote sustainability." (Kurp, 2008, pg. 11).

Forrester Research defines green IT as, *IT suppliers and their corporate customers changing the way computing assets are designed, manufactured, operated, and disposed of to gain efficiency and cost savings while reducing environmentally harmful impacts. Concerned about the environmental footprint of their companies, enterprises are constantly consolidating." (Staten, 2008, pg. 4).*

Economically, green computing strives to achieve economic viability as well as improved system performance and use while simultaneously abiding by our social and ethical responsibilities to contribute to a sustainable future. Thus, green IT must out of necessity include: the dimensions of economic sustainability; the economics of energy efficiency; and the total cost of ownership, including the cost of disposal and recycling.

Green IT is the study and practice of using computing resources efficiently across all of these dimensions. The goals of green computing include: reducing the use of hazardous materials; maximizing energy efficiency during the product's lifetime; and promoting recyclability and biodegradability of defunct products and factory waste (Murugesan, 2008).

Ruth (2009) believes that Green IT will become an integral part of the wider green movement because, "it is the largest single source of hazardous waste, or e-waste" and data center and server energy costs are substantial (pg. 78).

Importance to Business

According to the 2010 report on *The State of Green Business,* what began as an ostensibly benevolent venture, green business shifted to a way to cut costs and improve corporate reputations. According to Makower (2010), "it has become a fundamental business competency, alongside accounting, finance, human resources, marketing, customer service, procurement, knowledge management and others." (pg. 4).

In November of 2009, the Wall Street Journal published a quiz, *How Well Do You Know…Green IT?* (Bulkeley, 2009). It included some interesting "fast facts" on green computing:

- Zombie servers are those that keep running even though they have nothing to do. These servers use a significant amount of electricity
- The consumption of electricity used by storage devices grew 191% between 2000 and 2006
- Smart grid development efforts could create up to 280,000 jobs by 2012
- IBM lead Greenpeace's CoolIT Challenge because of their proactive role in reducing their own emissions and broad scope of climate solutions
- By turning off corporate computers at night, companies would save an estimated 20 million tons of carbon-dioxide emission in a year

Green IT has become a part of an elemental change in both the economy and society. According to Mines & Davis (2007), motivation for change came from government mandates, energy efficiency, product and company differentiations, and corporate social responsibility (CSR). In order for green computing and sustainability practices to work, they must be incorporated into the supply chain. Both suppliers and buyers understand that they must integrate green or sustainability practices into design, manufacturing, operation and disposal of their IT resources (Mines & Davis, 2007). Table 1 shows the evolution of sustainability practices; that is, it indicates how society currently participates in or views various sustainability issues, and how we will practice and view them in the future.

	The Present		**The Future**
Green regulations	Voluntary	→	Mandatory
Green consumers	Minority	→	Majority
Investor focus	Growing	→	Intense
Executive view	Leading-edge	→	Mainstream
Business initiatives	Niche projects	→	Core to business
Carbon tax	None	→	Multiple
Green IT	Interesting	→	Required

Source: Forrester Research, Inc. (2007)

Table 1: Predicted Changes in Sustainability Practices

Importance to Education

Given the overwhelming movement toward sustainability practices and toward the incorporation of green IT into the corporate enterprise, it is incumbent upon educators to prepare students for a green workforce. According to The College Student Educators International (ACPA), "There is an urgent need for US higher education to advocate for sustainable development." (*Education for Sustainable Development,* n.d., pg. 1).

The American College & University Presidents Climate Commitment (ACUPCC) contends that higher education plays a "unique and critical role, one often overlooked, in making a healthy, just and sustainable society and a stable climate a reality." (pg. 5). The ACUPCC also believes that colleges and universities will be challenged to meet their responsibility to "provide the knowledge and educated citizenry that will lead to a thriving civil society." (ACUPCC, 2009, pg. 5).

The United Nations declared a ***Decade of Education for Sustainable Development*** (2005-2014) to "promote the local and global acceptance of principles of 'sustainable development'." (Education for Sustainable Development, n.d., para 1). HEASC, the Higher Education Associations Sustainability Consortium (www.heasc.net), plans to champion sustainability education throughout institutions of higher education.

The Bureau of Labor statistics (2009) predicted that employment of environmental scientists was expected to grow by 25% between 2006 and 2016. Beginning in fiscal year 2010, the US Bureau of Labor Statistics (BLS) will work with other Department of Labor (DOL) agencies and key organizations to define and produce data on green jobs, http://www.bls.gov/green. There is also a website dedicated to Green Jobs, http://www.greenjobs.net.

Corporate IT organizations and their suppliers consider environmental responsibility to be an emerging critical topic in most organizations. Green IT is a fundamental ingredient of corporate social responsibility and is often a starting point for such efforts because it yields timely operational and cost benefits (McClean, 2008).

Sadly, according to the ACUPCC (2009), the vast majority of college graduates know little about the importance of climate change or sustainability. In addition, they do not have their "personal, professional and civic lives aligned with sustainability principles." (pg. 13). This is both a challenge and opportunity for higher education to move swiftly to prepare students for a green workforce. According to ACPA, higher education can change operational, curricular, and policy norms so students can become "environmentally responsible, socially responsible, economically responsible, and active citizens in a global economy." (Education for Sustainable Development, n.d.). (pg. 2).

Degree programs

The demand for Green IT knowledge by those in industry has yet to make its way to the academic environment. The Association for the Advancement of Sustainability in Higher Education (AASHE) indicates that as of April 2010, 34 colleges and universities are offering academic degrees in Sustainability – 17 offering sustainability degrees at the baccalaureate level, 15 offering sustainability degrees at the masters level, and 5 offering sustainability degrees at the doctoral level.

As of this writing it appears that there are no colleges and universities offering degrees in Green IT at any level, and our survey results indicate but one Swedish university offering a minor in Green IT (though this minor could not be verified on the university web site). Almost all of the minor programs are in sustainability, and none of the minor programs in sustainability include a course in Green IT.

A sustainability curriculum framework, *Education for Sustainability,* was developed by Second Nature (www.secondnature.org). Second Nature's mission is, "to accelerate movement toward a sustainable future by serving and supporting senior college and university leaders in making healthy, just, and sustainable living the foundation of all learning and practice in higher education." (Second Nature Mission Statement, para. 1).

There are seven critical sustainability themes within their curriculum framework, including *Technological and Economic Relationships to Sustainability.* The technology theme includes the following topical areas:

- Technical, scientific and institutional strategies that foster sustainable development
- Energy and natural resource efficiency and conservation
- Shifting from nonrenewable resources (e.g., fossil fuels) to renewable resources
- Prevention and control of pollution and waste
- Design for the environment, industrial ecology and ecologically sustainable design
- Remediation of current environmental problems and preservation of biological diversity

In the world of IT, there is plenty of potential for advancement for those who know their way around green issues. The Chief Green Officer position is one that IT professionals are ideally suited for (Wheeland, 2009).

It is no secret that there is plenty of work ahead of us in moving the U.S. and other nations to a green economy. The trouble is not in finding people who need work, but rather in finding qualified and well trained workers to take on those jobs (Terry-Cobo, 2009).

For companies facing a shortage of applicants, a promising solution is in the works: Green jobs training programs will make it easier for companies to make those hires, and federal, state and local projects are helping spread those nationwide. However, according to Terry-Cobo (2009), there is currently no concrete definition of "green-collar jobs".

2. METHODOLOGY

In spring 2010, the authors sent a request for participation to the academicians in higher educational organizations. There were 89 participants who replied from the USA and eight other countries.

Reliability

The survey data was analyzed with descriptive statistics, cross tabulations, and analysis of variance (ANOVA) using SPSS 17.0. To avoid research bias, ANOVA with post hoc analysis was also used to determine whether the statistically significant differences found between the independent and dependent variables were reliable (Field, 2000).

Instrument

A web-based survey which was composed of twenty-seven questions included both Likert scale and open-ended question types. The questions were then grouped into three parts to analyze the following categories: (a) Part I: Demographic Information (8 questions); (b) Part II: Courses Related to GITS areas (10 questions) (see Appendix B). After we received the human subject protection approvals from each institution, an email was sent out to several universities in the USA and in other countries. The survey link was activated from the end of January to mid-March 2010.

Research Hypotheses

The purpose of this study is related to Green Information Technology (IT), Green Computing, and/or Sustainability (GITS) curriculum initiatives in institutions of higher education. The purpose of this study is threefold: 1) to evaluate whether GITS academic programs have been initiated in higher educational organizations; 2) to analyze if GITS programs and/or courses are currently in place; and 3) to investigate the conceptual framework of GITS across campuses. Therefore, we proposed and tested the following hypotheses:

H1: GITS academic programs have been initiated in more than 50% of the higher educational organizations.
H2: GITS programs are currently in place at more than 50% of the higher educational organizations.
H3: There will be greater GITS course content in non-US versus US countries.

Participants

Participants were invited from the colleagues of professional organizations and two major international conferences: Information System Educators Conference and Science Mathematics Technology Educators. A total of 185 invitations were sent and 89 participants completed the survey; they included 25 (28.7%) from the US Northeast; 12 (13.8%) from the US Midwest; 24 (27.6%) from the US South, 5 (5.7%) from the US West, and 21 (24.1%) from a country other than the US. Of those, eight countries were included: Argentina, Britain, Germany, Hong Kong, Ireland, Malaysia, Sweden, and Taiwan. Fifty-four percent of the participants were from educational institutions that have a total student population of greater than 10,000 students. A total of 61 colleges and universities were identified in this study. Gall, Gall, and Borg (2003) stated that the minimal total sample sizes for independent samples t test with Alpha at the .05 level of significance is 64 based on the medium effect size. In this study, we collected 89 samples which met this criterion to provide a reliable analysis for a medium effect size.

3. FINDINGS

The findings provided below are based on the research hypothesis in the following section.

H1: GITS academic programs have been initiated in more than 50% of the higher educational organizations.

Our survey results indicated that a very high percentage of colleges/universities did not offer a GITS minor (89%) or GITS major (84%). One institution claimed to offer a degree in Green IT, and one institution claimed to offer a degree in Green Computing, but an extensive web search could not confirm these results. Further, one institution claimed to offer a minor in Green IT, and one institution claimed to offer a minor in Green Computing, but once again an extensive web search could not confirm these results. One possible explanation for this discrepancy is perhaps those answering the survey are doing so in anticipation of forthcoming programs that have yet to be approved and/or updated on the institutions' web sites. What we did find is that almost all of the minor programs are in sustainability, and that none of the minor programs currently include a course in Green IT.

Our results indicated that 30.6% of participants expect their college or university administrators to initiate and/or promote a Green IT or Green Computing academic program on campus. On the other hand, 27.4% of participants indicated that it was unlikely that their administration would do so. When asked about the expectation for the college/university faculty to design Green IT / Green Computing courses, 28.8% of the respondents indicated that they were likely to do so, but 42.4% of responses indicated the opposite expectation. Based on the analysis, we found that GITS academic programs have not been initiated in more than 50% of the higher educational organizations, therefore H1 was rejected.

H2: GITS academic programs are currently in place at more than 50% of the higher educational organizations.

We found that 82.7% of the responses indicated that neither Green IT nor Green Computing courses were offered by their college or university. There are only three institutions who offer the GITS course in both undergraduate (9.6%) and graduate levels (3.8%). However, there are four other institutions where a GITS course is currently under development. Within programs of the 89 respondents, only 17 (19.1%) indicated that Green IT is covered in any of their IS/IT courses. In response to a related question, 97% of those who responded cover Green IT either little or very little.

The survey results of this study indicated that 5 universities are currently offering a course in Green IT. However, a look at the courses listed on their respective web sites could not verify the survey results. At best we can conclude that these courses are currently being offered as special topics courses, that the topic of Green IT was being covered as a portion of another course, or perhaps they are still under development.

Overall, research hypothesis 2 cannot be supported. GITS programs are currently not in place in more than 50% of the higher educational organizations. Very few have them and very few spend significant time on GITS topics in their other coursework.

H3: There will be greater Green GITS course content in non-US versus US countries.

Many countries outside of the US are viewed as having higher Green initiatives. Therefore it was expected that the universities surveyed outside the US would have higher Green IT course content. A comparison of means test for difference in course content between US and other countries found the same mean for both samples as shown in Table 5 (see Appendix). Hypothesis 3 was rejected. No difference was found between the US and other countries with regard to current GITS content.

Summary of Findings

Among the suggested course topical areas regarding to GITS, we found that only a very small percent (less than 13.4%) of participants have very little Green IT course content. The feedback from the participants stated a common theme that higher education administrators need to be responsible for supporting sustainability curricula. These courses will provide the necessary foundational knowledge for students as future employees who enter the workforce having an understanding of the green economy and sustainability issues. Moreover, some comments stated that the time for GITS is now. Many educated people, including faculty and students, are unaware of the importance of sustainability issues and of the impact of Green IT efforts on the environment. Therefore, "awareness and enforcement" are needed for higher education to take for the first step.

While some participants stated that they were encouraged by having the vision of implementing GITS in their college/university, a few participants expressed their concerns regarding the practices in GITS. Among the concerns is the belief that their college/university administration is not willing to support GITS because they believe that GITS is not a "front burner" issue. It is unlikely to happen unless university administrators and faculty are able to see the advantages that Green IT could bring to the community and beyond. Some participants opined that although GITS is an important topic, it does not constitute a course. Some of the participants expressed the view that a "special topics course" might be sufficient or to offer Green IT as a standing elective course. One respondent stated, "Green IT is an ascendant concept in academia and will take some time to catch on."

4. RESULTS

Regarding the correlations among the participants' demographic information, courses related to GITS areas, and the sustainability initiative in the school, we summarize the following findings from this study.

The results for our proposed hypotheses are stated as that follows:

H1 was rejected. The GITS academic programs have not been initiated in more than 50% of higher educational organizations collected from this study.

H2 was rejected. The GITS academic programs are not currently in place at more than 50% of the higher educational organizations collected from this study.

H6 was rejected. There was no difference between US vs. non-US countries in amount of GITS coverage.

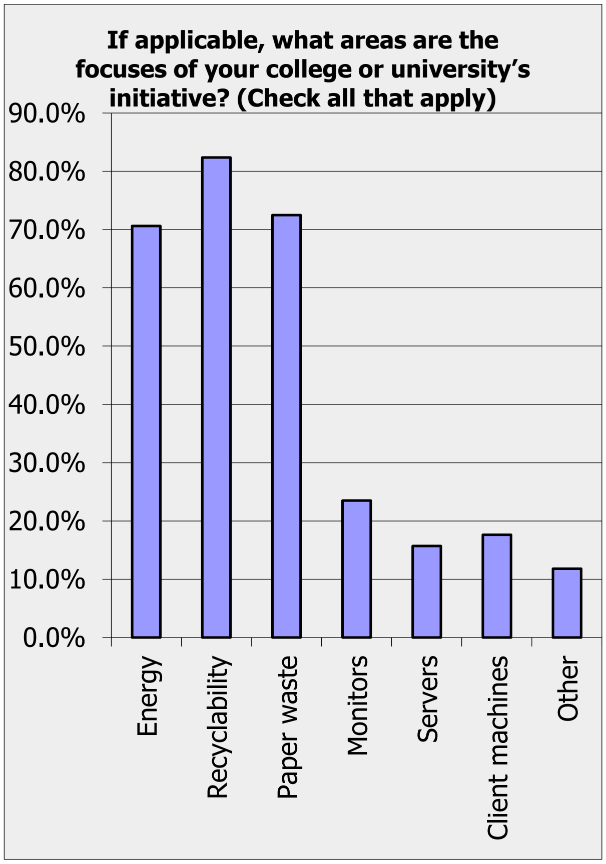


Figure 1. GITS Focuses

When asked what areas their respective college or university focused on with relation to GITS, we found the top three items that stood out were: Recyclability, Energy, and Paper Waste (see Figure 2). Overall, 56.7% of the respondents expect that Green IT/Green Computing courses will benefit environmental sustainability issues in the long run, but 17.9% expressed an unlikely expectation. To carry on the concern of the environmental sustainability issues, 55.2% of participants would be willing to assist with implementing Green IT/Green Computing courses in their department or college. Only 13.5% responded that they are unlikely to assist in this area; however, 31.3% were neutral.

Among the suggested GITS course topical areas, the respondents suggested the following areas, as indicated in Table 2, from the most to the least important:

Table 2. Suggested GITS course topical areas

GITS Topical Area	Frequencies
Recyclability	8
Energy	6
Paper Waste	6
Sustainability Studies	6
Operations and Infrastructure	5
Energy Management Systems	5
Budget Management	4
Client Machines	4
Data Center Energy Consumption	3
Server Virtualization and Consolidation	3
Servers	3
Monitors	1

Table 3. GITS topical areas currently taught in other courses

Topics Taught in other Courses	Frequencies
Energy Management Systems	9
Energy	7
Data Center Energy Consumption	7
Operations and Infrastructure	7
Recyclability	7
Server Virtualization & Consolidation	7
Client Machines	6
Paper Waste	6
Budget Management	5
Sustainability Studies	5
Servers	4
Monitors	2
Other—Environmental Management	1

Twenty-five percent of those responding answered that they covered GITS topics in the courses that they are currently teaching. The respondents indicated that the following topics, as indicated in Table 3, were covered in their courses from the most to the least popular. We also verified that 86.6% of the participants spent very little time on coursework that encompasses Green IT. GITS academic programs have not been initiated in more than 50% of higher educational institutions that participated in this study.

5. RECOMMENDATIONS

This study concluded that while there is an interest by higher education academicians in Green IT and Sustainability studies, very few institutions have incorporated Green IT into their curriculum. Based on the survey findings and a content analysis of several Green IT and sustainability books that are currently on the market (see Appendix A), we have recommended topical areas for a first course in Green IT (see Table 4). Interestingly, many of these areas mapped to the topical areas that many of the respondents have already incorporated into other classes in their IS/IT curriculum, or areas that the respondents perceived as notable.

While there are no textbooks in Green IT on the market to-date, Appendix A provides a list of Green IT and sustainability books (listed by publication date), websites, newsletters, and certification sites that one could utilize as resources and references in the Green IT course. Some of the books, e.g. *Greening through IT: Information Technology for Environmental Sustainability* and *Green IT for Dummies*, provide an overview of many of the areas listed in Table 3, while others, e.g., *The Shortcut Guide to Data Center Energy Efficiency*, provide information on very specific topical areas. In addition, one can find information on sustainability throughout in books such as, *Hot, Flat, and Crowded 2.0: Why We Need a Green Revolution--and How It Can Renew America.*

Incorporating the GreenBiz.com newsletters, *Greener Computing News and GreenBuzz*, into the Green IT course, will provide the students with up-to-date contemporary issues for discussion and research. While some sample videos were provided in Appendix A, it is recommended that faculty and students search YouTube at least once per semester for specific topical areas, e.g. e-waste, for access to current videos.

If the course does not have another sustainability course as a pre-requisite, we recommend that the Green IT course begins with an overview of sustainability concepts, followed by an overview of Green IT concepts. We recommend no specific order in which to present the other topical areas. While other subject areas were presented in the GITS books that were reviewed, the twelve recommended topics below were presented in two or more books. This list should provide the instructor a strong foundational course in Green Information Technology.

Table 4. Content analysis of Green IT publications: GITS topical areas

GITS Topical Areas	Frequency Ranking
Virtualization Virtual Servers Desktop Virtualization Server ROI & Implementation	1
Energy Use Carbon/Environmental Footprint Metrics Power Savings Energy Resource Efficiency	2
Technology Infrastructure Resource Management Green Hardware	3
Green Data Center Cooling Data replication Efficiency Design	4
Human Factors Employees/Virtual Workers Customers Creating a Green Culture	5
Document and Data Storage Management	6
Green IT Checklist Planning, Policies, Strategies Problems/Pitfalls	7
IT Asset Disposal & Recycling e-Waste	8
Green Computing & Sustainability Basics	9
Legal/Government Standards	10
Paperless	11
Green Supply Chain	12

6. CONCLUSIONS

The demand for Green IT knowledge by those in industry has yet to make its way to the academic environment. The Association for the Advancement of Sustainability in Higher Education (AASHE) indicates that as of April 2010, thirty-four colleges and universities are offering academic degrees in Sustainability – 17 offering sustainability degrees at the baccalaureate level, 15 offering sustainability degrees at the masters level, and 5 offering sustainability degrees at the doctoral level.

As of this writing it appears that there are no colleges and universities offering degrees in Green IT at any level, and our survey results indicate but one Swedish university offering a minor in Green IT (though this minor could not be verified on the university web site). Almost all of the minor programs are in sustainability, and none of the minor programs in sustainability include a course in Green IT.

Our survey results indicated that five universities are currently offering a course in Green IT. However, a look at the courses listed on their respective web sites could not verify the survey results. At best we can conclude that these courses are currently being offered as special topics courses, or that the topic of Green IT was being covered as a portion of another course.

EDUCAUSE concluded that engagement of most institutions of higher education in sustainability practices is "more optimistic than systemic." (Sheehan & Smith, 2010, pg. 18). Cameron (2009) stated, "As IT professionals, educators, and researchers, we share a responsibility to identify key contributors to energy and environmental waste in our field. But simply identifying problems is not enough; we also have the responsibility to act." Survey respondents shared similar expectations toward environmental sustainability issues.

Acccording to Jucker (2003), "Sustainability is achieved when all people on Earth can live well without compromising the quality of life for future generations." (pg. 85). Institutions of higher learning should therefore strive to explore the value of Green IT & Sustainability initiatives through academic programs to fulfill a common goal as digital citizens.

7. REFERENCES

Adams, W.M. and Jeanrenaud, S.J. (2008). *Transition to Sustainability: Towards a Humane and Diverse World.* IUCN: Gland, Switserland. <note: IUCN is the International Union for Conservation of Nature and Natural Resources>.

American College & University Presidents Climate Commitment. (2009). *Education for Climate Neutrality and Sustainability: Guidance for ACUPCC Institutions.* Boston: ACUPCC.

Bulkeley, W. M. (2009, November 16). *How Well Do You Know...Green IT?* Retrieved November 18, 2009, from The Wall Street Journal: http://online.wsj.com/article/SB10001424052748704224004574489620527547560.html

Cameron, K. W. (2009, January). Green Introspection. *Computer* , 101-102.

Choi, Y. B., & Mata-Toledo, R. (2009). *Green Computing.* Unpublished manuscript.

Education for Sustainability. (2010). Retrieved April 10, 2010, from Second Nature: http://www.secondnature.org/about

Education for Sustainable Development. (n.d.). Retrieved March 9, 2010, from ACPA - College Student Educators International: http://www.myacpa.org/task-force/sustainability/docs/SustainablePrimer.pdf

Field, A. (2000). *Discovering statistics using SPSS for Windows.* Thousand Oaks, CA: Sage Publications.

Gall, M.D., Gall, J.P., & Borg, W.R. (2003). Educational research an introduction. (7th ed.). Boston: Pearson Education.

Jucker, R. (2003). UNESCO Teaching and Learning for a Sustainable Future: A Critical Evaluation of Underlying Unsustainable Progress Myths. *Trumpeter , 19* (2), 83-107.

Kurp, P. (2008). Green Computing: Are You Ready for a Personal Energy Meter? *Communications of the ACM , 51* (10), 11-13.

Lamb, J. (2009). *The Greening of IT: How Companies Can Make a Difference for the Environment.* Boston, MA: Pearson, IBM Press.

Lyon, J., Sargent, R.-M., & Dorney, R. (2010). *Environmental Studies and Sustainability Business Plan.* Unpublished Report, Merrimack College, North Andover, Massachusetts.

Makower, J. (2009, February). *State of Green Business 2009.* Retrieved August 9, 2009, from GreenBiz.com: http://www.greenbiz.com/sites/default/files/StateofGreenBusiness2009.pdf

Makower, J. (2010). *The State of Green Business 2010*. Retrieved April 10, 2010, from GreenBiz.com: http://www.stateofgreenbusiness.com/

McClean, C. (2008). *Topic Overview: Corporate Social Responsibility.* Cambridge, MA: Forrester Research, Inc.

Mines, C., & Davis, E. (2007). *Topic Overview: Green IT.* Cambridge, MA: Forrester Research, Inc.

Murugesan, S. (January/February 2008). Harnessing Green IT: Principles and Practices. *IEEE IT Professional*, *10* (1), 24-33.

Rosenbaum, K. L. (1993). Sustainable Environmental Law: Integrating Natural Resource and Pollution Abatement Law from Resources to Recovery. In C. Campbell-Mohn (Ed.), *Environmental Law Institute* (pp. 575-674). St. Paul, MN, USA: West Publishing Co.

Ruth, S. (2009). Green IT--More Than a Three Percent Solution? *IEEE Internet Computing , 13* (4), 74-78.

Sheehan, M.C., & Smith, S.D. (2010, April). *Powering Down: Green IT in Higher Education.* EDUCAUSE Center for Applied Research. Retrieved June 10, 2010 from http://www.educause.edu/Resources/PoweringDownGreenITinHigherEdu/203028.

Staten, J. (2008). *Topic Overview: IT Consolidation.* Cambridge, MA: Forrester Research, Inc.

Terry-Cobo, S. (2009, July 6). *Green-Collar Graduations Show the Promise of Stimulus Funds*. Retrieved April 10, 2010, from GreenBiz.com: http://www.greenbiz.com/news/2009/07/06/green-collar-graduations-show-promise-stimulus-funds

Vieira, P. F. (1993). *As ciencia sociais e a questao ambiental: Rumo a Interdisciplinaridade.* Rio de Janeiro, Brazil: APED/NAEA/UFPA.

What is Sustainability? (n.d.). Retrieved April 4, 2010, from ASU Global Institute of Sustainability: http://sustainability.asu.edu/about/perspectives.php

Wheeland, M. (2009, August 25). *Green IT Degrees: Coming Soon to a School Near You*. Retrieved April 10, 2010, from GreenBiz.com: http://www.greenbiz.com/blog/2009/08/25/green-it-degrees-coming-soon-school-near-you

Editor's Note:
This paper was selected for inclusion in the journal as an ISECON 2010 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2010.

Appendices and Annexure

Table 5. Descriptive—Green IT Course Content in US vs. non-US Countries

GI Covered

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
US	52	1.75	.437	.061	1.63	1.87	1	2
Other Country	16	1.75	.447	.112	1.51	1.99	1	2
Total	68	1.75	.436	.053	1.64	1.86	1	2

Appendix A

Green IT and Sustainability Resources

BOOKS

Greening through IT: Information Technology for Environmental Sustainability
Bill Tomlinson
Hardcover: 216 pages
Publisher: The MIT Press (May 3, 2010)
ISBN-10: 0262013932
ISBN-13: 978-0262013932

The Shortcut Guide to Data Center Energy Efficiency
David Chernicoff
Published April 28, 2010
Format: Kindle Edition
File Size: 1169 KB
☐ Publisher: Realtime Publishers (March 1, 2010)
☐ ASIN: B003M68XP8
http://www.greenbiz.com/business/research/bookstore/2010/04/27/shortcut-guide-data-center-energy-efficiency#ixzz0qS38NALP

CompTIA Strata - Green IT Certification Exam Preparation Course in a Book
William Manning
Paperback: 188 pages
Publisher: Emereo Pty Ltd (January 15, 2010)
ISBN-10: 1742442463
ISBN-13: 978-1742442464

Hot, Flat, and Crowded 2.0: Why We Need a Green Revolution--and How It Can Renew America
Thomas L. Friedman
Paperback: 528 pages
Publisher: Picador; Exp Upd edition (November 24, 2009)
ISBN-10: 0312428928
ISBN-13: 978-0312428921

The Green Collar Economy: How One Solution Can Fix Our Two Biggest Problems
Van Jones
Paperback: 272 pages
Publisher: HarperOne (September 29, 2009)
ISBN-10: 0061650765
ASIN: B003GAN3FK

Grow a Greener Data Center
Douglas Alger
Paperback: 336 pages
Publisher: Cisco Press; 1 edition (August 27, 2009)
ISBN-10: 1587058138
ISBN-13: 978-1587058134

Foundations of Green IT: Consolidation, Virtualization, Efficiency, and ROI in the Data Center
Marty Poniatowski
Paperback: 352 pages
Publisher: Prentice Hall; 1 edition (August 21, 2009)
ISBN-10: 0137043759
ISBN-13: 978-0137043750

Green Recovery: Get Lean, Get Smart, and Emerge from the Downturn on Top
Andrew Winston
Hardcover: 200 pages
Publisher: Harvard Business Press (August 1, 2009)
ISBN-10: 1422166546
ISBN-13: 978-1422166543

Green Tech: How to Plan and Implement Sustainable IT Solutions
Lawrence Webber, Michael Wallace
Hardcover: 292 pages
Publisher: AMACOM; 1 edition (July 28, 2009)
ISBN-10: 081441446X
ISBN-13: 978-0814414460

The Greening of IT: How Companies Can Make a Difference for the Environment
John Lamb
Paperback: 352 pages
Publisher: IBM Press; 1 edition (May 7, 2009)
ISBN-10: 0137150830
ISBN-13: 978-0137150830

The Green IT Pocket Pack
By View Site
Published April 28, 2009

The complete set includes:
The Green Agenda: A Business Guide (Softcover)
The Green Office: A Business Guide (Softcover)
Compliance for Green IT: Pocket Guide (Softcover)
http://www.greenbiz.com/business/research/bookstore/2009/04/28/green-it-pocket-pack#ixzz0qS54nvEV

Green IT For Dummies
Carol Baroudi, Jeffrey Hill, Arnold Reinhold, Jhana Senxian
Paperback: 364 pages
Publisher: For Dummies (April 27, 2009)
ISBN-10: 0470386886
ISBN-13: 978-0470386880

Green to Gold: How Smart Companies Use Environmental Strategy to Innovate, Create Value, and Build Competitive Advantage
Daniel Esty, Andrew Winston
Paperback: 408 pages
Publisher: Wiley; Rev Upd edition (January 9, 2009)
ISBN-10: 0470393742
ISBN-13: 978-0470393741

The Green and Virtual Data Center
Greg Schulz
Hardcover: 396 pages
Publisher: CRC/Auerbach Publications; 1 edition (January 26, 2009)
ISBN-10: 1420086669
ISBN-13: 978-1420086669

Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line
Toby Velte, Anthony Velte, Robert Elsenpeter
Paperback: 308 pages
Publisher: McGraw-Hill Osborne Media; 1 edition (September 8, 2008)
ISBN-10: 0071599231
ISBN-13: 978-0071599238

75 Green Businesses You Can Start to Make Money and Make A Difference
Glenn Croston
Paperback: 328 pages
Publisher: Entrepreneur Press; 1 edition (August 1, 2008)
ISBN-10: 1599181800
ISBN-13: 978-1599181806

ENCYCLOPEDIAS

Green Business: An A-to-Z Guide
Nevin Cohen, general editor/Paul Robbins, series editor
Electronic ISBN 978-1-4129-7379-3
2010 by SAGE Publications, Inc.

CERTIFICATION

Green IT Certification
Global Knowledge
http://www.globalknowledge.com/training/course.asp?pageid=9&courseid=13290

Tonex Green IT Boot Camp
http://www.tonex.com/Courses/100103/

CompTIA Strata Green IT Certificate
http://www.comptia.org/certifications/strata/greenit.aspx

MSP Alliance Green IT Certification
https://www.mspalliance.com/why-mspa/green-it-certification-program

CISCO Green IT Certification
https://learningnetwork.cisco.com/community/learning_center/green_it

VIDEOS

Green Recovery: Emerge from the Downturn on Top
http://www.youtube.com/watch?v=WLX_hKT3JTY
HarvardBusiness — August 25, 2009 — Andrew Winston, founder of Winston Eco-Strategies, explains how to get lean, get smart, and emerge from the downturn on top. 12 min

Green Recovery: Get Lean, Get Smart, and Emerge from the Downturn on Top
http://www.youtube.com/watch?v=EC_l6TdCSI4
August 28, 2009 — Andrew Winston is interviewed by Harvard Business Publishing on Green Recovery. 4:11 mins

Explaining Green Computing
http://www.youtube.com/watch?v=350Rb2sOc3U
August 22, 2008 — When it comes to being green, computing is both part of the problem and part of the solution. This video therefore looks at the environmental aspects of computing, and is presented by Christopher Barnatt, author of ExplainingComputers.com, and Associate Professor of Computing and Organizations in Nottingham University Business School. 7:18 mins

5 Green Computing Tips
http://www.youtube.com/watch?v=SdxPgjfQ9yk
March 17, 2008 — InformationWeek Green Computing blogger Cora Nucci offers advice to make your data center more eco-friendly, and economical, too. 2:29 mins

The definitive guide to *green computing*
http://www.computing.co.uk/computing/video/2224462/ctg-def-guide-green
Jul 10, 2008 - Rising energy costs mean going *green* makes business sense - we look at the latest trends. 6 min

Earth Week Green Computing Solutions Series
e.g., Earth Week: Green Computing Solutions #2 - Top Green Web Sites
http://www.watchdoit.com/how-to-videos/Earth-Week-Green-Computing-Solutions-2-Top-Green-Web-Sites-36946.aspx

WEBSITES & NEWSLETTERS

Greener Computing News and GreenBuzz e-newsletters
www.greenbiz.com
Newsletter comes once per week, includes links to blogs and podcasts

Greener Design
www.greenerdesign.com

Green Life
www.thegreenlife.org

Second Nature
www.secondnature.org

Sustainability
www.sustainability.com

Village Earth
www.villageearth.org

Appendix B

Survey

Part I: Demographics

1. What is the total student population of your college or university (your location only)?
   Less than 1000
   1000-2000
   2000-5000
   5000-10000
   +10000

2. Where is your school located?
   US Northeast
   US MidWest
   US South
   US West
   Other country __________

3. What type of IT degree do you offer?
   CS
   IS
   IT
   MIS
   None
   Other _________

4. What is the gender type of your IT students?
   < 10% Female
   10-25% Female
   25-50% Female
   More than 50% Female

5. What is your gender?
   Female
   Male
   Prefer not to answer

6. What is your age range?
   18-25
   25-35
   35-50
   50+
   Prefer not to answer

7. What is your academic rank?
   Adjunct
   Instructor
   Assistant Professor
   Associate Professor
   Professor
   Other _________________

8. What is the name of your university? (This information will be kept in the strictest confidence and will not be used in the report manuscript)

9. What is the name of your department? (This information will be kept in the strictest confidence and will not be used in the report manuscript)

Part II: Green IT / Green Computing courses

1. Does your university offer a Green IT, Green Computing or Sustainability major? (Check all that apply)

Green IT
Green Computing
Sustainability
None of the above
Currently under development
Other______________________

2. Does your university offer a Green IT, Green Computing or Sustainability minor? (Check all that apply)

Green IT
Green Computing
Sustainability
None of the above
Currently under development
Other______________________

3. Does your university offer a Green IT or Green Computing course? (Check all that appy)

Green IT
Green Computing
Both
Neither
Currently under development
Other___________________

4. If so, which topical areas are included or proposed? (Check all that apply)

Budget Management
Client machines
Data Center Energy Consumption
Operations and infrastructure
Energy
Energy Management System
Monitors
Paper waste
Recyclability
Server virtualization and Consolidation
Servers
Sustainability Study
None
Other _____________

5. If so, at which academic level? (Check all that apply)

Undergraduate
Graduate
Both UG and Grad
Do not offer

6. Do you personally offer this course?

Yes
No, another colleague teaches this course
Do not offer

7. If your university offers neither a Green IT nor a Green Computing course, are Green IT, Green Computing, or Sustainability covered in any of your IS/IT courses?

Yes
No

8. If so, which topic areas are covered? (Check all that apply)I

Budget Management
Client machines
Data Center Energy Consumption
Operations and infrastructure
Energy
Energy Management System
Monitors
Paper waste
Recyclability
Server virtualization and Consolidation
Servers
Sustainability Study
None
Other _____________

9. If so, would you mind sharing a copy of your syllabus with us?

Yes
No
If yes, what is your email address so that we might contact you:
____

10. On a scale of 1 to 5 (1-very little, 5- a great deal), how much of your coursework deals with Green IT?

Determining the Most Suitable E-Learning Delivery Mode for TUT Students

Solomon Adeyemi Odunaike,
Odunaikesa@tut.ac.za;

Daniel Chuene
ChueneND@tut.ac.za

Department of Software Development,
Tshwane University of Technology (TUT)
Pretoria, 0001 South Africa

Abstract

Traditionally, in education and business environment, Information Technology has been seen as purely a support or operational tool. Advances in computing, information storage, software, and networking are all leading to new tools for teaching and learning and are also changing the paradigm for new initiative in the classroom teaching. The Internet Technology is in forefront of transforming education and opportunities around the world by allowing different kind of interaction and innovation among various educational institutes and students alike all participating in the global online innovations. The new modes of learning need to be explored to determine possible impact and most suitable choice of strategies. Understanding our student extents and capabilities of learning on their own coupled with the availability of basic and necessary equipments required for e-learning will have profound impact on the choice of e-teaching, e-learning and e-education delivery mode. A right choice of delivery mode is very essential and fundamental in moving forward to attain the much needed greater height in quality education. A wrong choice however, could deal a devastating blow on any progress made so far in the implementation of e-learning which would in turn have a ripple effect on the quality, acceptability level, pass rate and cost. Every effort must be made to guard against making wrong choice of e-learning delivering mode and this research will investigate and evaluates which e-learning delivery mode will be best suitable for Tshwane University of Technology (TUT) students.

Keywords: E-learning, E-education, Internet technology, On-line education

1. INTRODUCTION

Schools, institutions of higher learning, corporate and industry leaders have discovered the impact of multimedia presentations and how it can aid in delivering their message. It is an undeniable fact that Internet Technology is taking the lead in transforming education and thereby providing limitless opportunities around the globe for educational institutions and students alike to collaborate irrespective of where they are located. E-learning is one such opportunity provided for by internet technology. It can be defined broadly as a way using technology to deliver learning and training programs. Arguably, a wrong choice of e-learning delivery mode could deal a devastating blow on any progress made so far on the implementation of e-learning which would in turn have a ripple effect on the quality, acceptability level, pass rate and cost. Every effort must be made to guard against making wrong choice of e-learning delivery modes, and this leads to our research question:

Which e-learning delivery mode will be best suitable for TUT students?

TUT is a University of Technology based in the Gauteng Province in South Africa. It was established in 2004 out of a merger of three former Technikons namely, Technikon Northern Gauteng (TNG), Technikon North West (TNW) and Technikon Pretoria. It has three main campuses in the Gauteng Province, and four satellite campuses in other provinces. It enrols approximately 60 000 students which are drawn from the nine provinces in South Africa as well as neighbouring countries like Botawana, Namibia, Zimbabwe and Swaziland. A large component of the current student population is from rural South Africa.

The main aim of this research is to determine the most suitable e-learning delivery mode for TUT students. Furthermore and towards answering our research question, the following section 2 will highlight and present the details of e-learning, the benefits thereof and types of e-learning delivery mode. Section 3 will discuss the research methodology. Finally, section 4 will present the findings of this research while section 5 will conclude the study. This section will summarise and make informed conclusion and recommendations on our choice of e-learning delivery mode and the future e-learning education in our institutions.

2. LITERATURE REVIEW

Advances in Information Technology have brought about various new tools that facilitate teaching and learning. Varieties of new modes of learning are also emerging and are pushing for new initiatives in classroom teaching and learning. (Rosenburg, 2001) mentions online learning as one viable mode of instruction. He however indicates that there are several issues that need to be taken into consideration. One is that online training packages should not merely be a replication of stand-up training; but the structure of the delivery mode should take into cognisance the needs of the learner as well as the learning situation of the learner. It should be borne in mind that is not all institutions of higher learning that can manage to offer learning by digital means due to various reasons. (Rosenburg, 2001) cites readiness and willingness to share information, the will by management to invest in infrastructure and resources, the readiness of trainers to design curriculum as key factors that need to be considered in order for deliver by digital means to succeed.

E-Learning Education

It is very common these days to hear arguments here and there that instructional technology in form of e-initiatives will be the key to educational quality in the new millennium. (Fiske & Hammond 1997) argue that quality education is a universal goal. This view was equally supported by (Mergendollar, 1996) which implores and urges policy makers to encourage a greater investment in educational technology as a means and path to educational quality. Other educational technology enthusiasts like (Connick, 1997) argue that educational technology in form of e-learning initiatives will dominate the environment for a long time to come with outstanding quality and will continue to increase rapidly to create "new educational culture".

(Campbell, 2001) defines e-education to involve e-teaching and e-learning along with various administrative and strategic measures needed to support teaching and learning in an Internet environment. Furthermore e-education could be further defined as the delivery of education by electronic means using formal teaching methodologies and styles.

E-Learning delivery mode

It was observed by (Kapur, 2003) that several web sites and education companies are offering Web-based solutions and that many are turning to the net for education. While online courses are most of the time strictly online, delivery of e-learning will include amongst others additional features such as live chats between students and lecturers, online assignments, discussion boards, playback of recorded material and email support. The key benefit with e-learning is that unlike with normal regular classes, students can learn at their own pace. With features that allow for the recording of classes, missed classes will be a thing of the past. Technology-enhanced education covers a wide range of teaching and training:

- Using internet as electronic repository (where learning material and course are posted on the Learning Management System (LMS))
- On-line or e-mail enhanced courses (those taught in a regular classroom and

supplemented with email chats or content delivery - hybrid)
- Comprehensive online courses (those taught 100 percent online using a variety of Internet technologies).
- Using Blogs to frequently and chronologically publish personal thoughts.
- Podcasts allows a subscriber to receive programs and can listen to them at leisure. They are generally audio in MP3 format; however other formats such as video can be podcasted as well.

Reasons for E-Learning

The Internet as a powerful new means of communication is global, fast and growing rapidly making the world seemingly smaller and more connected, transmitting information at nearly real-time speed. No doubt, the World Wide Web is bringing rapid, clinical and radical changes into our lives. For Education, the efforts are very glaring. The Internet is making it possible for more individual than ever before to access knowledge and to learn in new and different ways. This not withstanding; (Rosenburg, 2001) infers there are reasons why the use of web in the classroom has not been gaining the required momentum and not been widespread. The reasons include but not limited to the followings:

- Knowledge – While it is not very simple and straightforward to create and maintain an extensive Web-based instructional site, other learners cannot manage themselves and learning on their own
- Reluctance – some educator are hesitant to adopt new methods or techniques of instruction while learners are battle ready to resist any form of change (culture of resistance).
- Resources – few schools can afford the time, support, training and recognition for teachers who wish to pursue new methods of instruction as demanded by Web-based instruction. Equally it is very likely that majority of learner can afford and maintain on-line connection
- Infrastructure – some schools lack the resources to develop large computer infrastructure as demanded by Web-based instructor.

Despite all the above negative impacts, there are numerous reasons for wanting to implement a Web-Based Instruction in a traditional classroom environment. Aldrich & Ross (2000) suggest that for one obvious reason, most educators would implement Web-Based Instructor with the aim of using a teaching method that is effective, efficient and enjoyable. Other advantages include the following:

- Enhancing student learning
- Spending more time with the students working in a small group or one-on-one
- Reducing repetitive teaching task
- Reducing paper flow and management
- Providing improved instruction materials

3. RESEARCH METHODOLOGY

Sampling method

Conforming to (Dix, Finlay, Abowd, & Beale, 1998), who state that the best way to find out how a system meets users' requirement and expectation is to 'ask the user'; it is therefore very important that we receive feedback from various users regarding their choice of e-learning technology.

Survey was used as the query technique to obtain information from the respondents. The questionnaire is targeted at the student currently being taught in the classroom and also using some WebCT to complement their studies as well as the lecturer who uses these applications in performing their duties.

Survey Population

The population for this survey are the students of TUT who are currently being taught in the traditional way and have at their disposal WebCT for their basic educational needs. Also included in this survey are lecturing staff that use WebCT and / or classroom environment to deliver lectures, together with the support staff charged with system / fault maintenance. The population in this study are mostly second level students (undergraduates) who are studying towards a diploma and Btech students (post diploma) studying Information and Communication Technology.

Leedy & Ormrod,(2001) observed that rather than sampling a large number of people with the intent of making generalizations, qualitative research tends to select few participants who can best shed light on the phenomenon or study under consideration. Therefore, the above suggests that convenient sampling of the population is highly favoured and would be more appropriate in this situation.

Survey data collection and analysis

The scalar style of questions was used. This is an adoption of the Likert technique. (Corbetta, 2003) stated that (Likert, 1932) proposed the scaling at the beginning of 1930s. The style expects the user to judge a specific statement on numeric scale of 1 to 5, usually corresponding to a measure of agreement or disagreement and may be in ascending or descending order of importance.

Although, open-ended questions give users an unrestricted or unbounded option of stating the issues the way they see it. It is not used in this study except where users' opinion is sought. Also, a "Yes" or "No' question was included to get affirmative answer from the respondent.

The analysis for the survey data will be done by collating all the responses. Pictorial analysis using chart will also be used.

4. ANALYSIS OF THE RESULTS

Analyzing the most suitable E-Learning delivering mode for TUT students

Before Internet access and web based applications became widely available, traditional classroom education use cable TV, video tapes, chalk and boards, magic markers and poster paper, tape recorded, movie projector, film strip projector, slide projectors, overhead projector, VCRs and a host of other equipments to deliver instruction to the learner. (Arsham, H. 2002) noted that each time there is a major and significant change in educational media, each time enthusiasts have announced with keen interest the transformation or even the end of the school system. Very rampant these days are talks of compromised education quality.

We believe that these technological innovations are profoundly influencing traditional university practices and policies and may even be fundamentally altering our conceptualizations of education. Hence the needs to find and evaluates the best suitable model.

Detailed Survey Results per Question

The survey instrument we used is shown in Appendix A. One hundred and sixty (160) questionnaires were sent out; one hundred and forty four (144) representing 90% responses were received. Hence the computation and evaluation will therefore be based on the 144 returned responses. All the sections of the questionnaire will be discussed and analyzed separately. Inferences will also be drawn on each of the questions.

Analysis of Survey Results in Section A

The objective of the grouping in this section is to ensure that the questionnaire was evenly distributed among students and gender classification so that the impending results could attest to the general feelings all groups that cut across gender, personnel and level of study.

Gender	Respondent	%
Male	91	63
Female	53	37
Undecided	0	0

Table 4.1: Tabulated survey results summarizing gender distribution

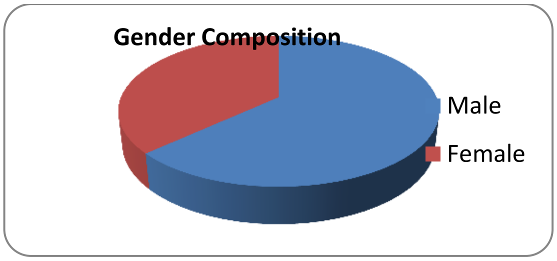


Figure 4.1: Graphical survey results summarizing gender distribution

Respondents Group	Respondent	%
Lecturers / Admin	18	13
Students	126	87
Undecided	0	0

Table 4.2: Tabulated survey results showing personnel distribution of questionnaire

The results show that the questionnaire was fairly distributed among the gender to reflect the gender composition of our classroom (2:1). The impending results from this study would reflect the general feelings that surpass issues of gender inequalities.

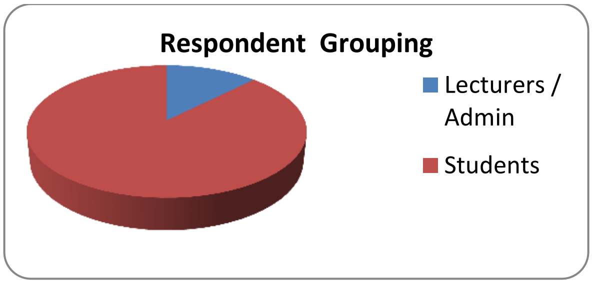


Figure 4.2: Graphical survey results depicting the distribution of questionnaire among respondents

We could not get more staff to fill the questionnaire because the timing of distributing the questionnaire coincided with examination periods and marking – a hectic period in academic calendar in terms of work load. Otherwise, there was a positive response from the students and few lecturers.

Level	Respondent	%
Diploma	47	33
BTech	94	65
MTech	0	0
Undecided	3	2

Table 4.3: Tabulated result showing specialization responses to the questionnaire

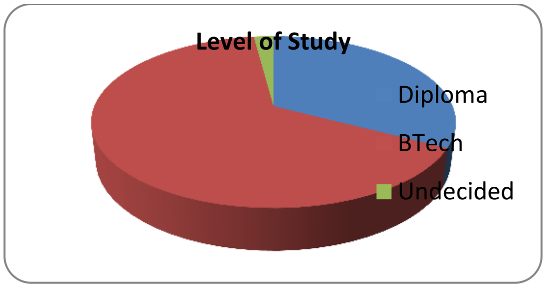


Figure 4.3: Graphical survey result depicting different specialization responses to the questionnaire

The Btech students were specifically included in this survey so that we could tap into their experience in seeking the best delivery mode for e-learning because most of them are working and could afford basic requirements of e-learning.

Analysis of Survey Results in Section B

The main reason of questions in this group is to evaluate the availability of computer and Internet requirements for e-learning. This is necessary because majority of our students are from rural areas where some of the equipment needed for e-learning may be luxury.

Have computers?	**Respondent**	**%**
Yes	125	87
No	17	12
Undecided	2	1

Table 4.4: Tabulated survey results showing respondents owning computer / laptop**.**

It is worth noting that 87% of the respondents have a computer / laptop. This is in sharp contrast of the study carried out by (Dehinbo, J. O. 2002) which indicated that only 15% of the students have access to computer and subsequent research carried out by (Odunaike S.A. 2006) which reflects 63%

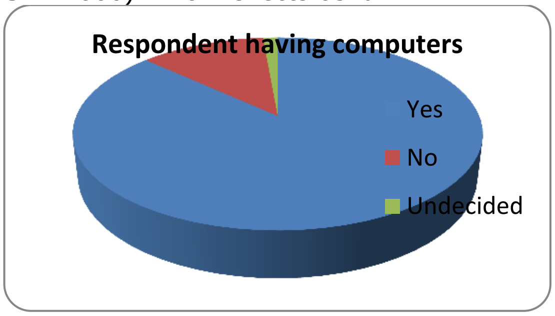


Figure 4.4: Graphical survey result showing respondents owning computer / laptop.

This means that students are have begun to realise the importance of having a computer which in turn frees up space for other students in the computer facilities provided by the school.

	Yes	**No**	**Undecided**
Respondent	122	20	2
%	85	14	1

Table 4.5: Tabulated survey results showing availability of computer during school hours.

The majority of the respondents indicated that they have their own computers, we expect the few respondents to make use of the computer laboratory and facilities provided by the institution during school hours.

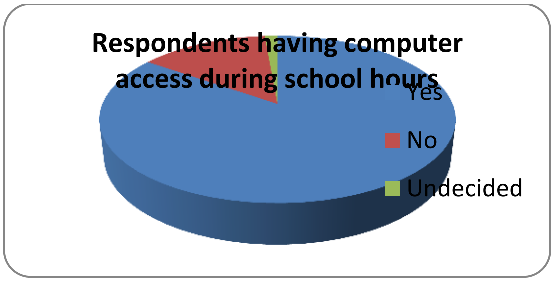


Figure 4.5: Graphical survey result depicting availability of computer during hours.

	Respondent	**%**
Yes	126	88
No	17	11
Undecided	1	1

Table 4.6: Tabulated survey results showing availability of computer after hours.

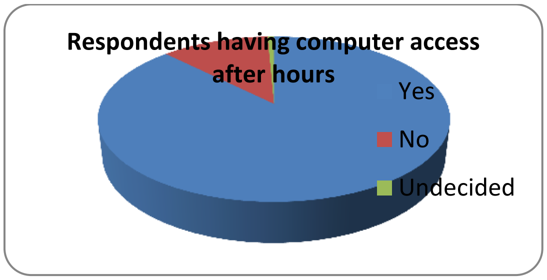


Figure 4.6: Graphical survey result showing the availability of computer after hours

The total of 126 respondents representing about 88% of the population has access to computer facilities after hours. An encouraging number indeed, efforts should be made to communicate the afterhours open laboratory facilities for the other group of students struggling to have afterhours access.

	Yes	No	Undecided
Respondent	109	32	3
%	76	22	2

Table 4.7: Tabulated survey results showing availability of Internet facility during hours.

The majority representing about 76% of the respondents have access to Internet facility during schools hours. The same number of respondents was not having internet access some years ago in the study conducted by (Odunaike S.A 2006). It means efforts of the Faculty of ICT to provide a dedicated Laboratory equipped with internet access is paying dividends.

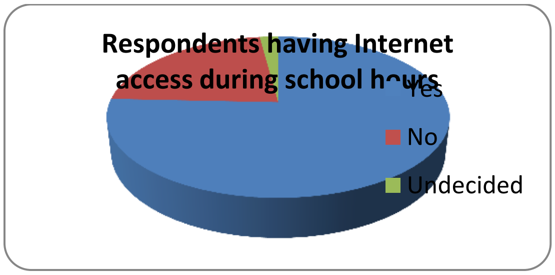


Figure 4.7: Graphical survey result depicting availability of Internet facility during hours.

	Yes	No	Undecided
Respondent	100	43	1
%	69	30	1

Table 4.8: Tabulated survey results showing availability of Internet facility after school hours.

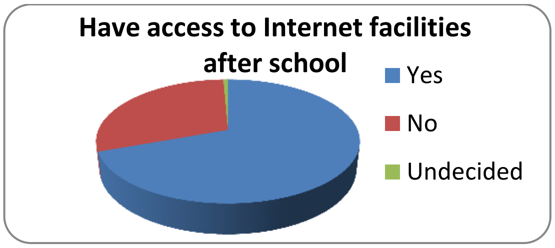


Figure 4.8: Graphical survey result depicting availability of Internet facility after hours.

Like B4, the respondent's 70% is very encouraging and highly welcomed as a factor in determining the best suitable e-learning delivery model. It is worth noting that the earlier research revealed the same amount of respondent was not having access couple of years back.

Accessing Internet after hours?	Respondent	%
Home	13	9
Work / School	90	63
Internet Café	24	17
Mobile Phone	27	19
3G	25	17
Undecided	2	1

Table 4.9: Tabulated results showing where respondent access Internet facility during hours.

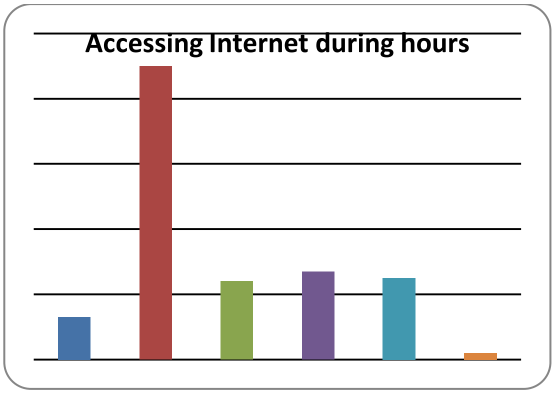


Figure 4.9: Graphical survey result depicting places of accessing of Internet facility during hours.

The majority of the respondents, (63%) rely heavily on their respective work place and school for internet connection. There could be only one reason for this phenomenon, avoiding high cost of internet connection. Also, the trend these days is to make connection through mobile phone and 3G (36%) at a budgeted rate.

Accessing Internet after hours?	**Respondent**	**%**
Home	15	10
Work / School	17	12
Internet Café	29	20
Mobile Phone	49	33
3G	44	31
Undecided	5	3

Table 4.10: Tabulated results showing where respondent access Internet facility after school hours.

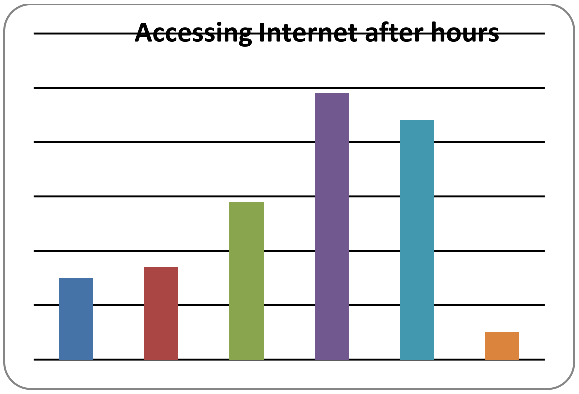


Figure 4.10: Graphical survey result depicting places of accessing of Internet facility after hours.

Most of the Internet access afterhours takes place using mobile phones and 3G (64%). On the other hand, respondents are willing to stay behind at work or school to get on-line connection. Others relied on friend or go to public place like the library or post office and internet cafe to access but just for very limited hours or on weekends probably.

Service	***Speed (Mbps)***	***Usage (GB)***	***Price (ZAR)***
Vodacom HSDPA	1.8	1	289.00
Sentech	1	a	299.00
Telkom ADSL	0.375	1	310.00
MTN	-	-	-
iBurst	1	1.2	369
Neotel	-	-	-

Table 4.11: Broadband costs per month
Source:jump.co.za

There are six main internet providers in South Africa; they are MTN, Vodacom, Neotel, iBurst, Sentech, and Telkom. Their internet connection rates vary as indicated below. **($1 = R7.35).**

Cost of Internet access	**Respondent**	**%**
R100 – R200	55	38
R201 – R300	26	18
R301 – R500	23	16
R501– R700	8	6
R701 and above	5	3
Undecided	24	17

Table 4.11: Tabulated results showing cost of access to Internet facility by respondents.

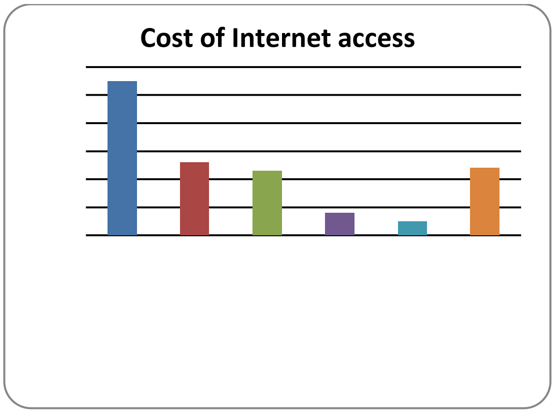


Figure 4.11: Graphical survey result depicting cost of accessing of Internet facility by respondents.

Most of the Internet access afterhours take place using Internet cafés, mobile phones and 3Gs. It becomes evident that most respondents budget as little as R100 for internet access. In our opinion, most internet access and download will be done either in the office or on the school dedicated laboratory.

Affordability	**Yes**	**No**	**Undecided**
Respondent	58	79	7
%	40	55	5

Table 4.12: Tabulated survey results showing affordability of on-line connection.

About 79 representing 55% respondents are not able to bear the cost while 5% are undecided. These figures give insight into the current cost of call and inter-connection rates which are high. We commend the efforts of ICT faculty for the dedicated laboratory and propose that more of such labs should be made available.

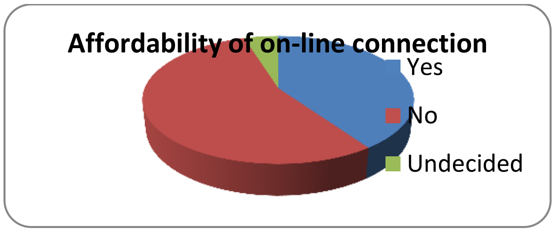


Figure 4.12: Graphical survey result depicting affordability of on-line connection.

In table 13, respondents make on-line learning their preferred choice by a 71% margin. This development is welcomed as it shows e-learning acceptability level in our institution.

Choice	Yes	No	Undecided
Respondent	102	40	2
%	71	28	1

Table 4.13: Tabulated survey results showing choice of on-line study.

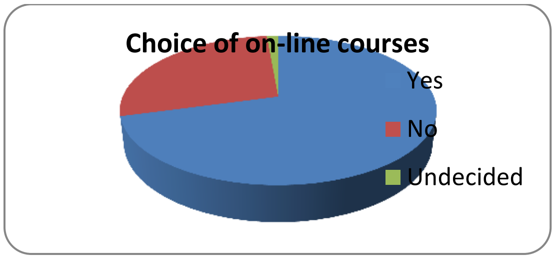


Figure 4.13: Graphical survey result depicting choice of on-line study.

Analysis of Survey Results in Section C

The main reason of questions in this group is to evaluate e-learning acceptability level, and easiness in using internet for educational and research purposes. More importantly, this section is used to evaluate the respondents' previous experience and ability to manage their study in internet education. It becomes necessary to evaluate their readiness for e-learning.

	Respondent	%
Strongly Agree	77	54
Agree	46	32
Undecided	12	8
Disagree	2	1
Strongly disagree	2	1
No response	5	4

Table 4.14: Tabulated survey results showing ease of using internet for education purpose.

From the responses in table 4.14, it shows that 86% are able to use computers and internet for education and research purposes while the remaining 14% either disagree or has no opinion on the issue.

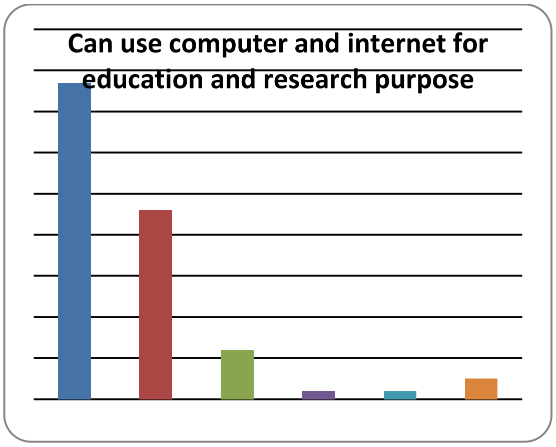


Figure 4.14: Graphical survey result depicting the ease of using the utility program.

Table 4.15 shows that there was a 96% response rate to this question and about 42% are actually in favour of this question while 32% fall short of finding on-line education without class attendance convenient. The result are not convincing, we are of the opinion that the department embark on awareness drive to gather support about e-learning education.

	Respondent	%
Strongly Agree	30	20
Agree	31	22
Undecided	31	22
Disagree	29	20
Strongly disagree	17	12
No response	6	4

Table 4.15: Tabulated survey results on user conveniences of on-line education, No class attendance.

	Respondent	%
Strongly Agree	26	18
Agree	31	22
Undecided	39	27
Disagree	23	16
Strongly disagree	20	14
No response	5	3

Table 4.16: Tabulated survey responses showing confidence in passing on-line education.

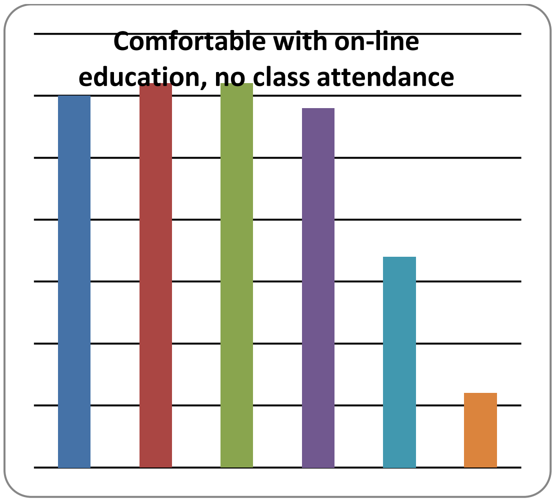


Figure 4.15: Graphical survey result on user conveniences of on-line education, No class attendance.

	Respondent	%
Strongly Agree	26	18
Agree	31	22
Undecided	39	27
Disagree	23	16
Strongly disagree	20	14
No response	5	3

Table 4.16: Tabulated survey responses showing confidence in passing on-line education.

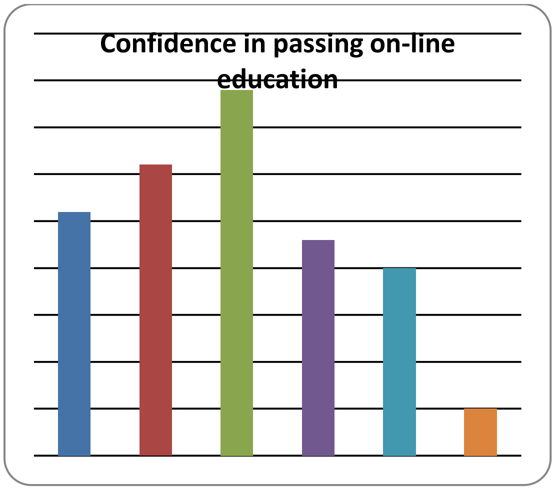


Figure 4.16: Graphical survey result depicting confidence in passing on-line education.

Despite the 97% response rate to this question, a below average rate of 40% have confidence in passing on-line courses while 30% respondents cannot guarantee they will pass; the other 30% are undecided. We agreed that the no classroom contact as stipulated by question generated some panic.

	Respondent	%
Strongly Agree	19	13
Agree	22	15
Undecided	33	23
Disagree	31	22
Strongly disagree	32	22
No response	7	5

Table 4.17: Tabulated survey responses showing respondents experience in on-line education.

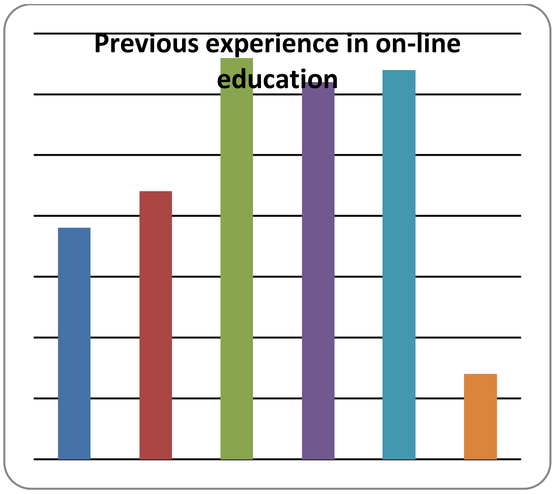


Figure 4.17: Graphical survey result depicting their experience in on-line education.

The response rate was 95%, a low proportion of 28% respondents have previous experience in on-line education and about 72% have no on-line education experience or undecided. This highlights that crops of students have no previous experiences with education technology but experience technology as they progresses with their studies.

	Respondent	%
Strongly Agree	33	23
Agree	43	30
Undecided	31	22
Disagree	21	15
Strongly disagree	10	7
No response	6	4

Table 4.18: Tabulated survey responses showing flexibility of on-line education.

About 53% of the respondents believe that on-line education is users-friendly and flexible. While 22% disagree on this issue and 25% remains undecided. We agree that based on the responses, respondents are prepared to attend courses at their own planning, space and time.

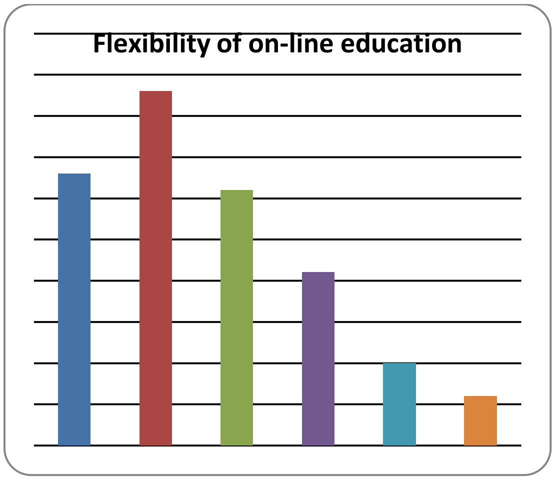


Figure 4.18: Graphical survey result depicting their flexibility of on-line education.

	Respondent	
Strongly Agree	23	16
Agree	26	18
Undecided	32	22
Disagree	28	19
Strongly disagree	29	20
No response	6	4

Table 4.19: Tabulated survey responses showing respondents choice of on-line education.

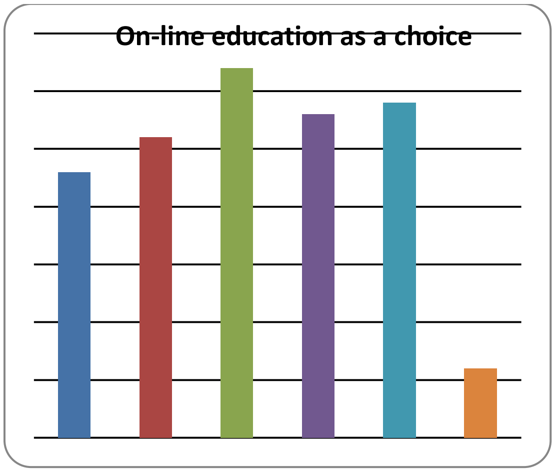


Figure 4.19: Graphical survey result depicting respondents' choice of on-line education.

The responses to choice of on-line education were no different from others. About 40% disagree on the choice of on-line education as a model and less than 35% prefer it as their choice. In our opinion, this has nothing to do with on-line education as a model but highlight the fear and lack of confidence among students to study on their own.

Analysis of Survey Results in Section D

The section was to ride on the back of the previous section C questions to get an affirmative answer and re-confirm the responses provided in the previous section regarding the choice of preference model of study.

	Yes	No	Undecided
Respondent	49	81	14
%	34	56	10

Table 4.20: Tabulated survey results showing preference of on-line education

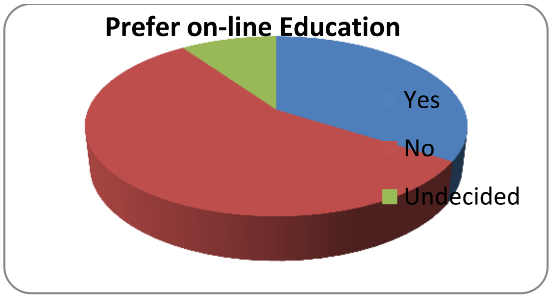


Figure 4.20: Graphical survey result depicting preference of on-line education.

The respondents re-affirmed their disagreement and non-preference of purely on-line education as e-learning delivery mode. There was no discrepancy from the previous and similar question in section C6. 34% against 56% respondents reportedly prefer online education.

	Yes	No	Undecided
Respondent	110	18	16
%	76	13	11

Table 4.21: Tabulated survey results showing choice of on-line combined with class education.

A clear margin of 76% are actually in favour of combing on-line education with classroom attendance while a mere 13% are against adopting on-line education with class attendance. The students made a statement of not abandoning traditional classroom education but we at least go for the combination. We suggest that the on-line education model should be introduced to them in stages.

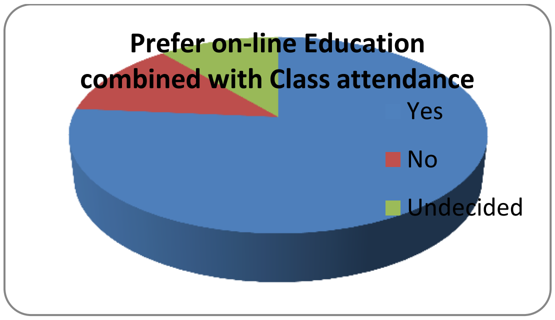


Figure 4.21: Graphical survey result depicting choice of on-line combined with class education.

A clear margin of 76% are actually in favour of combing on-line education with classroom attendance while a mere 13% are against adopting on-line education with class attendance. The students made a statement of not abandoning traditional classroom education but we at least go for the combination. We suggest that the on-line education model should be introduced to them in stages.

	Yes	No	Undecided
Respondent	61	63	20
%	42	44	14

Table 4.22: Tabulated survey results of on-line combined Technology with No class attendance.

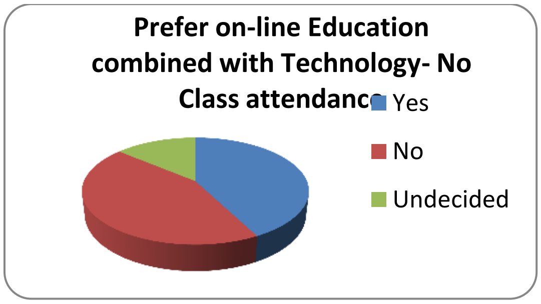


Figure 4.22: Graphical survey depicting on-line combined Technology with No class attendance.

The response to this question was a balanced one. About 42% are in favour of combing on-line education technology with various technologies but no classroom attendance whereas the same margins of 44% are against this form of learning model.

5. CONCLUSION

(Pajo & Wallace, 2001) reiterate that the growth in computer applications and the use of Internet in particular has signalled change for the delivery of education especially in the area of teaching and learning. Arguably, the use of this model in classrooms will be a major development that will change the way knowledge is imparted to students inside and outside the classroom. This is not the case in our institution; our learners are more comfortable with blended form of on-line education with classroom teaching.

The students should be encouraged to embrace technology in their learning by introducing them to educational technology as they progresses into their studies. We propose that one course at a time be delivered using on-line technology; while the other courses take turns to be implemented as on-line courses. We believe with time, the student will gain more confidence and enthusiasm for on-line education

The efforts of the Faculty of ICT in providing dedicated facilities are highly appreciated. The fact that more students (87%); could afford their own computers and have access to internet connection either through ICT facility or through the phone proves that the majority of the student populace are e-learning ready.

We noted that mobile phones have made internet connection more accessible and readily available. This is one technology that needs to be explored for flexible combination with on-line education.

Apart from being reluctance to accept changes – which could improve with time, inability to manage their study, inexperience in on-line education, we can conclude that the knowledge is there, the equipment is available and students can sustain on-line connection with infrastructure and resources provided by ICT management.

6. RECOMMENDATIONS

From the above findings and conclusion, it is evident that the learners at TUT will be comfortable with blended form of on-line education. We recommended that TUT should embark on a drive to make learners aware of the benefits that they can gain from e-education. We further recommend e-education be implemented with a pilot project and enhanced with various on-line technologies that can supplement classroom education.

7. REFERENCES:

Aldrich, C. (2000). The e-learning 2000 hype cycle. Stamford, CT. Gartner Group

Aldrich, C., & Ross, C. (2000). Evaluating an end-to-end e-learning infrastructural provider. Stamford, CT. Gartner Group

Campbell, N. (2001). e-teaching, e-learning and e-education: A paper to inform the development of the ICT strategy in New Zealand for Ministry of Education.

Connick, G.P. (1997). Issues and Trends to take us into the twenty-first century. New Directions for Teaching and Learning

Corbetta, P. (2003). Social Research: Theory, Methods and Techniques. U.K.: Sage Publications

Dehinbo, J. O. (2002). Frame work for determining suitable web development language for teaching in tertiary institution

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). Human-Computer Interaction. 2nd edition. U.K.: Prentice Hall

Fiske, E., & Bruce, H. (1997). Identifying quality in American colleges and Universities. Planning for Higher Education.

Franklin, S., & Peat, M. (2001). Managing change: The use of mixed delivery modes to increase learning opportunities. Australian Journal of Educational Technology, 2001 17(1),37-49

Jump.co.za (2007). Broadband price comparison http://www.jump.co.za/articles/2007/08/broadband-price-comparison.html Date accessed: 31/07/2010

Kapur, R (2003). Cyber Classroom www.rediff.com/search/2003/jan/22edu.htm Date accessed: 31/05/2010

Leedy, P.D., & Ormrod, J.E. (2001). Practical Research: Planning and Design. 7th edition. USA : Merrill Prentice Hall

Mergendollar, J.R. (1996). Moving form technologically possibility to richer student learning: Revitalizing infrastructure and reconstructed pedagogy. Section 4: Grading the policymakers' solution. Educational Researcher.

Odunaike, S. A. (2006). The Impact of incorporating e-learning in traditional classroom education

Pajo, K. and Wallace, C. (2001). Barriers To The Uptake of Web-based Technology by University Teachers.

Rosenburg, M. (2001). E-learning: strategies for delivering knowledge in the digital age. New York McGraw-Hill

Appendix A

Research Questionnaire

Determining the most suitable E-Learning delivering mode for TUT students

JUNE, 2010.

Brief Introduction

The current trend in society – globalisation, technology, life-long learning requirement are pressurising the existing models of education, learning content development and more importantly, the mode of delivering the instruction. Traditionally, in education and business environment, Information Technology has been seen as purely a support or operational tool. But since the advent of Internet and later electronic commerce and lately e-learning, there has been a shift in paradigm because the later has facilitated a more cost-effective way of doing things.

New approaches to learning may cause intense disruption to traditional higher education institutions. Hence new philosophies of learning need to be explored to determine the possible impact and choice of strategies. The views of students, lecturers and support personnel, learning outcomes, courses or module availability and delivery mode need to be evaluated and explored for relevance in the new learning climate.

Understanding our student extents and capabilities of learning on their own coupled with the availability of basic and necessary equipments required for e-learning will have profound impact on the choice of e-teaching, e-learning and e-education delivery mode. A right choice of delivery mode is very essential and fundamental in moving forward to attain the much needed greater height in quality education and making greater use of the opportunities provided by the Internet technology. This research questionnaire key focus and emphasis is on the changes to teaching and learning that will result from an e-education or on-line Education environment.

Please kindly complete this questionnaire based on your intuition, knowledge, personal experience with Internet, web-based education and your expectation. Your participation is highly welcome. Thank you.

S. A. Odunaike
Tshwane University of Technology
Faculty of Information and Communication Technology,
Soshanguve Campus.

SECTION A – contains sectional information to be used for grouping in order to obtain heterogeneous composition. Please mark the appropriate category with an X

A1	Gender	Male			Female	
A2	Technicians / Academic Assistant / Lecturers					
	Student					
A3	Present level of study		**NDIP**	**BTech**	**MTech**	

SECTION B – contains information to determine the availability of computer and internet – a basic requirement of E-Learning. Please mark the appropriate category with an X

B1	Do you have your own computer / laptop?		**YES**	**NO**		
B2	Do you have access to computer during school / work hour?		**YES**	**NO**		
B3	Do you have access to computer after school / work hour?		**YES**	**NO**		
B4	Do you access to Internet facility during school / work hour?		**YES**	**NO**		
B5	Do you access to Internet facility after school work hour?		**YES**	**NO**		
B6	Where do you get access to internet during school / work hours?	**Home Fixed line**	**Work School**	**Internet Café**	**Mobile Phone**	**3G**
B7	Where do you get access to internet after school / work hours?	**Home Fixed line**	**Work School**	**Internet Café**	**Mobile Phone**	**3G**
B8	If connection is by fixed line dial up / Internet Café / Mobile Phone / 3G, How much in Rand does it cost you?	**R100 - R200**	**R201 - R300**	**R301 - R500**	**R501 - R700**	**R701 and above**
B9	Can afford the cost of on-line connection every month?		**YES**	**NO**		
B10	Would you like some of your course / subject to be purely on-line / internet based?		**YES**	**NO**		

©2011 EDSIG (Education Special Interest Group of the AITP)

SECTION C – contains information to determine the level of acceptability and easiness in using internet and attending on-line courses. Please mark the appropriate category with an X

		5 Strongly agree	**4 Agree**	**3 Undecided**	**2 Disagree**	**1 Strongly Disagree**
C1	I find it easy to use the computer and internet for research and educational purpose.					
C2	On-line class is very convenient model to use for the course without class attendance					
C3	I have much confidence in myself using the on-line model and passing without Lecturer assistance.					
C4	I have done some correspondence or on-line course before. Don't think it's a problem.					
C5	The on-line model will be user friendly and flexible to allow me more time to study on my own plan, space and time.					
C6	I would prefer using the on-line / internet model for my course than attending classes.					
	Recommendation					

SECTION D – contains summary / concluding information to determine the preference model. Please mark the appropriate category with an X

D1	I prefer purely an on-line / internet model of learning.		**YES**	**NO**		
D2	I prefer on-line / internet model combined with class attendance and lectures.		**YES**	**NO**		
D3	I prefer on-line / internet model supplemented with on-line materials like slide, video, messaging, on-line mobile chat facilities, e.t.c. but NO CLASS ATTENDANCE and LECTURE.		**YES**	**NO**		

Thank you for your time and co-operation.

Beyond Introductory Programming: Success Factors for Advanced Programming

Arthur Hoskey
arthur.hoskey@farmingdale.edu

Paula San Millan Maurino
paula.maurino@farmingdale.edu

Computer Systems Department,
Farmingdale State College
State University of New York
Farmingdale, NY 11735

Abstract

Numerous studies document high drop-out and failure rates for students in computer programming classes. Studies show that even when some students pass programming classes, they still do not know how to program. Many factors have been considered to explain this problem including gender, age, prior programming experience, major, math background, personal attributes, and the programming language itself. Research in this area has mainly been confined to introductory programming courses. This study explores the problem at a higher level. It tracks students longitudinally as they move from the first introductory programming class, to the second introductory class, and finally, to completion of an advanced programming course. The research question answered was: What are the factors contributing to the success or lack of success in advanced programming? The success factors examined were the introductory programming language taken, number of programming classes taken, track (concentration in the major), math and logic background, time lapse between the introductory and advanced programming class, instructor, gender, and general GPA. The factors that influenced student success were found to be the introductory programming language, time lapse between the introductory and advanced class, general grade point average, and track. Identification of these factors will help educators to make the best decisions on how to improve computer curriculum and programs and help students become better programmers.

Keywords: programming, programming languages, programming success, programming failure, success factors

1. INTRODUCTION

Farmingdale State College, a campus of the State University of New York, is a four year college specializing in applied science and technology. The college has had in place a Bachelor of Science Degree in Computer Programming and Information Systems for the past eight years. The degree is offered by the Computer Systems Department in the School of Business at the college and has five tracks (concentrations within the major): networking, database, systems, programming, and web development. All students are required to take two semesters of programming at an introductory level. They are currently offered a choice of C++ or Visual Basic. In addition, they are all required to take an additional upper level programming course in Java. All students must achieve a "C" or better in both introductory programming classes to enter the advanced Java class.

Professors teaching the advanced course have found that some students entering the advanced class do not have the entry level programming skills needed to succeed in the upper-level class. Many possible explanations have been offered for this problem. It has been suggested by some faculty members that students wait too long to take the advanced course and as a result, have forgotten what they learned in the introductory classes. Others state that it is difficult for students to switch languages and recommend that all three courses use the same language. Still others state it is the introductory language that is at fault. They feel that Visual Basic is not an appropriate language for teaching programming and should be dropped from the curriculum or offered only as an elective. Some wonder if the fact that students do not do well in the required math courses or put off taking them could be related. Finally, others state that only students in the programming track do well in the course. Perhaps students in the other tracks should not have to take the advanced course.

This study was an exploration of this problem. We wanted to identify the factors involved in the apparent loss or lack of programming ability experienced by some students as well as the factors leading to success for others. Once these factors are identified, we will be able to make the best decisions on how to improve the program and help our students become better programmers. As such, our research question was: What are the factors contributing to the success or lack of success in advanced programming?

2. LITERATURE REVIEW

Failure/Drop Out Rates

As we searched the literature, we immediately realized we were not alone. Numerous studies document high drop out and failure rates for programming students (Guzdial & Soloway, 2002; McKinney & Denton, 2004). In a worldwide study, Bennedsen & Caspersen (2005) found that 33% of students fail CS1. Compounding the problem, some students pass, but do not actually learn to program. In a multi-national, multi-institutional study of assessment of programming skills of first year CS students, students averaged only 22.89 out of a possible expected 110 points (McCracken, Kolikant, Almstrum, Laxer, Diaz, Thomas, Guzdial, Utting, Hagan, & Wilusz, 2001). In a later study that built on the McCracken work, it was found that many students lacked the knowledge and skills that are a precursor to problem solving. They cannot read or systematically analyze a short piece of code (Lister, Adams, Fitzgerald, Fone, Hamer, Lindholm, Mc Cartney, Mostrom, Sanders, Seppala, Simon & Thomas, 2004).

Introductory Programming

Most of the literature in this area was confined to studying the problems encountered by students in introductory classes. The students in our research study have already completed two semesters of computing. Yet, some of these students appear to have the "shallow and superficial skills" described in a 2005 study of novice programmers by Lewandowski, Gutschow, McCartney, Sanders, & Shinners-Kennedy. In an international study of 500 students and teachers, Lahtinen, Ala-Mutka, & Jervinen (2005) found that the biggest problem of novice programmers is not the understanding of basic concepts, but rather learning to apply them.

Math/Prior Programming Experience

Many studies seeking to predict achievement in introductory programming courses have examined math background, previous programming experience, and previous academic background. Previous experience with programming and a math background seem to be positively related to success in introductory programming (Byrne & Lyons, 2001; Bennedsen & Casperson, 2005; Wilson & Shrock, 2001; Rountree, Rountree, Robins & Hannah, 2004). Once again, our students have completed two semesters of programming already. They are required to take calculus, but this is not a prerequisite for any of the programming classes. Some students procrastinate and put it off. Others need math remedial classes and cannot take it until those courses have been completed.

Other Personal Attributes

Some studies have looked at factors such as sex and age. These demographics do not seem to affect success in programming although the numbers of females entering programming is much lower (Bennedsen & Caspersen, 2005; Byrne & Lyons, 2001; Wilson & Shrock, 2001). Other studies have attempted to link programming success with a student's grades in previous coursework, self-efficacy, "comfort level" , or motivation to get an "A" in the course (Wilson & Shrock, 2001; Bennedsen & Caspersen, 2005; Rountree, Rountree, & Robins, 2001; Wiedenbeck, 2005).

Programming Language

Other studies looked at the programming language used in the classroom. Of these, some analyzed the languages for their teaching efficacy (Mannila, Peltomaki, & Salakoski, 2006; Mannila & de Raadt, 2006; Chen, Monge, & Simon, 2006; Dehinbo, 2006; Russell, Russell, Pollacia & Tastle, 2009; McIver & Conway, 1996) and others looked at the reasons colleges selected a particular language (Parker, Chao, Ottaway & Chang, 2006; Bhatnager, 2009).

There was no consensus on the best language to use. Lahtinen, Ala-Mutka, & Jervinen (2005) found that the teaching language did not seem to affect the learning situation. Chen, Monge, & Simon (2006) concurred. However, McIver & Conway (1996) found that a substantial part of the difficulty encountered in programming classes arises from the structure, syntax, and semantics of the particular programming language used. Further, Mannila, Peltomaki & Salakoski (2006) found that students did just as well learning a simple language and then moving on to a more complex one. They also found that the best languages to use in teaching programming were the languages designed with teaching in mind. They agreed with other researchers, however, that language is selected for many reasons beyond pedagogical benefit. In a study of employers and educators by Bhatnagar (2009), the teaching of more than one language was recommended.

Major

Lastly, some studies looked at the student's major. Prasad & Li (2004) tried to determine if there were differences between students majoring in computing and those majoring in information systems enrolled in the same computer programming course. They noted that information systems students had a little more difficulty with C++, but that the difference was slight. A student's major or intended major was found to be insignificant in a study done by Bennedsen & Caspersen (2005). Rountree, Rountree & Robbins (2001) found no difference in success rates for 472 students in an introductory programming class in Java for computer science majors, information science majors, or non-computer majors.

3. METHODOLOGY

Farmingdale State College's school records were used to create a database containing information about all two hundred students who took Java Programming from 2005 through the fall 2009 semester. After the statistical analysis for the years 2005-2009 was complete, we added the results for the spring 2010 semester. The spring 2010 Java class included 25 additional grades. The final database contained two hundred and twenty-five grades for Java. These final grades constituted our measure of success in the class. The statistical analysis was performed on the database of two hundred and twenty-five students unless indicated otherwise.

The database held information on each student in the following areas:

- The programming language taken in the introductory classes
- Whether or not a logic class was taken before the introductory programming class
- The number of programming classes taken
- Grades in the programming classes
- Overall GPA
- Time elapsed between the introductory programming classes and the Java class
- The particular professors teaching the programming classes
- Major or track (concentration within the Computer Systems Department)
- The type and sequence of math courses taken

Statistical analysis was performed on the data to determine relationships, if any, between the variables and student success in the advanced Java course. As mentioned previously, success in the Java course was measured by the student's final grade. In particular, we wanted to determine the following

- Did addition of a logic course to the curriculum increase success in programming?
- Did the particular faculty member teaching the introductory course affect student success in the advanced Java course?
- Was there a difference in male and female success rates in the Java class?

- Were students who took more than the minimum number of programming courses more successful in the advanced Java course?
- Did taking the required calculus course before Java increase success?
- Did the amount of time lapsed between taking the advanced Java course and completion of the introductory courses affect success in the Java class?
- Did students who took Visual Basic in the introductory courses do better or worse in the advanced Java class than students who took C++?
- Did students with a higher general GPA achieve greater success in the Java class?
- Did students in the programming track perform better in Java than students in the systems, web development and networking tracks?

4. RESULTS

Overview

A summary of our results appears in table 1 below.

Table 1: Summary of Study Results

Independent Variable	**Difference In Java Grades?**
Time Lapse Since Programming 2	Yes
Introductory Programming Language	Yes
Track (Concentration)	Yes
General GPA	Yes
Logic Course	No
Major	No
Faculty	No
Gender	No
Number of Programming Courses Taken	No
Math Courses Taken	No

The independent variables that produced a difference in the Java grades were: time lapsed since Programming 2, the introductory language taken, the track (concentration within the Computer Systems Department) taken, and general GPA (grade point average). The variables that did not produce a difference in the Java grades were: taking a logic course first, major, the particular faculty member that taught the introductory class, gender, number of programming classes taken, and math courses taken.

Time Lapse since Programming 2

Students who took Java the semester following the last introductory programming course had a higher mean average in the Java class than students who waited two or three semesters to take the course. The longer the time lapse, the more the more the mean average declined. See figure 1.

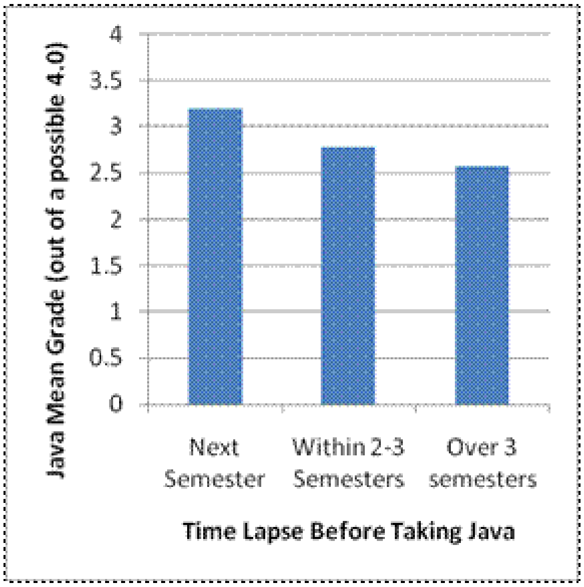


Figure 1: Time lapse between Programming 2 and Java and mean averages in Java

Along these same lines, the longer students put off taking Java after completion of Programming 2, the more likely they were to get below a 2.0 ("D" or "F") in the Java class. Of the students who took Java the following semester after Programming 2, 10% earned a "D" or "F" (under a 2.0 out of a possible 4.0). Twenty percent of students who waited two to three semesters to take Java after Programming 2, received a grade of "D" or "F". Twenty-two percent of students who waited over three semesters received a grade of "D" or "F". See figure 2.

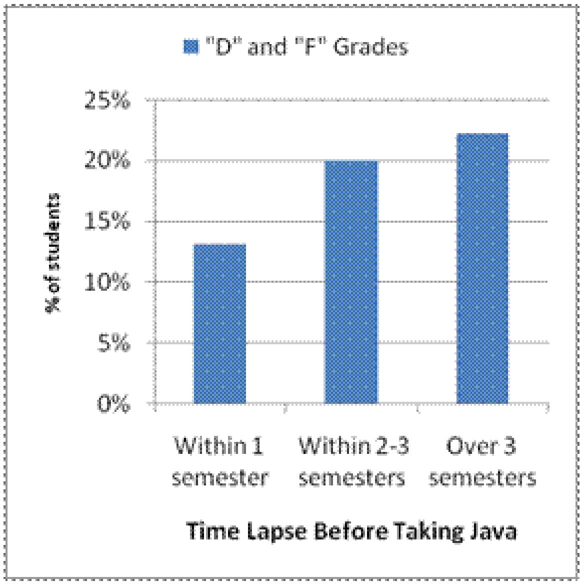


Figure 2: The number of "D" and "F" grades increase when students postpone taking Java.

The statistical validity of these findings was tested using a one-tailed Mann Whitney U Test. There was a significant statistical difference when *next semester* and *within 2-3 semesters* were compared. See table 2.

Table 2: Mean Averages of Java students Grouped by Time Lapse of Next Semester vs. Within 2 or 3 Semesters

	Next Semester	Within 2-3 Semesters
Mean	**3.19**	**2.79**
N	61	75
U = 1884		
Significance= p<.05		

Table 3: Mean Averages of Java Students Grouped by Time Lapse of Next Semester vs. Over 3 Semesters

	Next Semester	Over 3 Semesters
Mean	**3.19**	**2.58**
N	61	45
U = 969.5		
Significance= p<.01		

There was a highly significant difference when *next semester* and *over 3 semesters* was compared. See table 3.

Table 4: Mean Averages of Java Students Grouped by Time Lapse of Next Semester vs. Over 1 Semester

	Next Semester	Over 1 Semester
Mean	**3.19**	**2.71**
N	61	120
U = 2853.5		
Significance= p<.01		

The next semester mean was also compared to the average means for all students who waited over one semester and that result was found very significant. See table 4.

When means for a time lapse of one, two or three semesters were compared to over three semesters that was also found statistically significant. See table 5.

Table 5: Mean Averages of Java Students Grouped by Time Lapse of 1, 2, or 3 Semesters vs. Over 3 Semesters

	1, 2 or 3 Semesters	Over 3 Semesters
Mean	**2.97**	**2.58**
N	136	45
U = 2441.5		
Significance= p<.05		

The only comparison where a statistical significance was *not* found was when *two to three semesters* was compared to *three semesters*.

Introductory Programming Language

Table 6: Mean Averages of Java Students Grouped by Introductory Programming Language Taken from 2005-2009

	C++	VB
Mean	**2.80**	**2.13**
N	76	28
U = 818		
Significance= p<.05		

It was found that students who took C++ for introductory programming classes were more successful than students who took Visual Basic for introductory programming classes using a one-tailed Mann-Whitney U test.

C++ students in the 2005-2009 group attained an average grade of 2.80 on a 4.0 scale in Java. Visual Basic students in the 2005-2009 group attained a 2.13 grade in Java.

When this data was added to the spring 2010 semester, there was little difference. The C++ average was then 2.78 and the Visual Basic average 2.0. See table 7.

Table 7: Mean Averages of Java Students Grouped by Introductory Programming Language Taken from 2005-2010

	C++	VB
Mean	**2.78**	**2.20**
N	90	33
U = 1189		
Significance= p<.05		

Track/Major

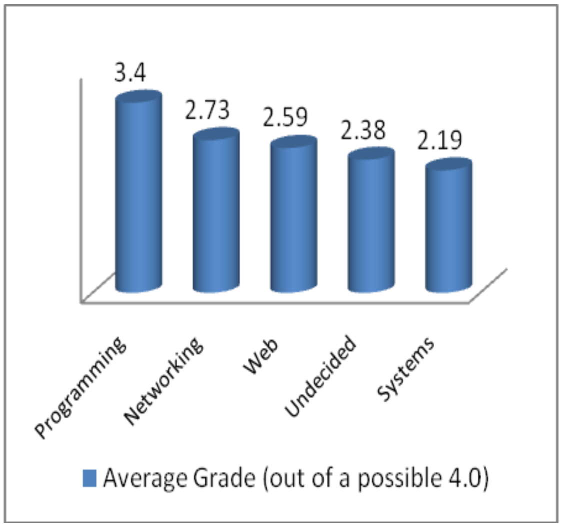


Figure 3: Success in the Java course by track

The Computer Systems Department has five tracks (concentrations) in a particular area. Each student selects one track and completes four courses in that area in addition to taking the other required courses in the curriculum. The two introductory programming courses and the advanced Java course are part of the core required curriculum, not a particular track. The five tracks are programming, web development, networking, systems and database. The database track was added last semester and as a result, was not considered in this research study.

It was found that students in the programming track were most successful in the Java course, followed by networking, web development, undecided, and systems. See figure 3 below which shows average means on a 4.0 scale for the four tracks and students who were undecided.

This difference was found to be highly significant using both a two-tailed Mann-Whitney U Test and a one tailed Mann-Whitney U Test. See tables 8 and 9 below.

Table 8: Comparison of Programming Track vs. **Not** Programming Track – Java Means

	Programming Track	**Not** Programming Track
Mean	**3.4**	**2.53**
N	45	180
U = 2579		
Significance = p<.01		

Table 9: Comparison of Programming Track and Other Tracks - Java Means

	Prog.	Net.	Sys.	Web Dev.	Undecided
Mean	**3.4**	**2.73**	**2.19**	**2.62**	**2.38**
N	45	56	29	50	32
U		927.5	298.5	701	445.5
Significance		p<.01	p<.01	p<.05	p<.01

The average mean of the systems track students was then compared to the average mean of all other tracks combined using a one-tailed Mann Whitney U Test. The findings were found significant at p<.01. See table 10.

Occasionally, students from outside the department take the Java class as an elective. Some of the other majors that have taken this course are nursing, bioscience, applied mathematics, and computer engineering. Also, it is taken infrequently by non-matriculated students who do not have a major. There was no significant difference found between the Computer Systems majors and non-majors.

Table 10: Comparison of Systems vs. Other Tracks – Java Means

	Systems	All Other Tracks
Mean	**2.19**	**2.78**
N	29	196
U = 3622.5		
Significance = p<.01		

General GPA

The student's general GPA average in the semester before the student took the Java course was compared to the grade the student earned in the Java class. A highly significant correlation was found between the student's general GPA and the Java grade using the Pearson product moment correlation coefficient (n=225, df=223, r = .52, p<.0005).

Logic Course

In an effort to improve performance in its programming classes, the department changed its requirements a few years ago to include a mandatory programming logic course. This logic course must be taken before the first programming class. No significant statistical difference was found between students who did or did not take the logic class before entering the first programming class.

Faculty

To determine if the particular faculty member teaching the introductory courses affected student success in the advanced Java course, we broke down the Java classes into groups based on the particular instructor that taught the introductory level class. No significant statistical difference was found in the final Java grades based on the faculty member who taught the introductory programming courses.

Gender

Females constituted only 11.60% of the students in the Java classes. Their mean average in the Java class was 2.66 out of a possible 4.0. Males in the Java courses (88.39%) had a mean average score of 2.7 out of a possible 4.0. Thus, no significant difference was found based on gender.

Number of Programming Courses Taken

The college offers a number of additional programming courses that are not required and can be taken as electives. Also, students may take C++ in the introductory courses and Visual Basic as an elective or vice versa. No significant statistical difference was found in the final Java grades for students who took more programming courses than required.

Math Courses Taken

Students are required to take two mathematics courses, Calculus and Methods in Operation Research. These math courses are not prerequisites for the Java course. It was found that there was no significant statistical difference between students who took Calculus before the advanced Java course and students who took calculus after the Java Course.

5. DISCUSSION

Based on the literature review, we expected to find that students who completed the newly required logic class, took Calculus before Java, and completed more programming classes than required would be more successful in advanced Java than students who did not. These factors, however, were all found to be statistically insignificant for our students.

It is surmised that the logic course may help prepare the students for programming, but not actually increase their programming ability. Anecdotally, instructors in the early introductory classes have stated that it is easier to teach programming to students after completion of the logic class. The instructors found that moving the material covered in the logic course out of the introductory programming course allowed them to devote more time to programming and gave them more time to cover all the required material. Thus, the course still appears to have value and will most likely be maintained in the curriculum.

It appears that the additional programming courses taken by some students did not help them succeed in the advanced Java course. Possibly these additional courses only serve to reinforce and reiterate material already covered. Another explanation might be that students may have difficulty transferring the skills from one language to another. A more accurate and comprehensive exploration of this issue will be undertaken in stage two of this research study. Stage two will use a qualitative approach with in-depth student interviews.

As stated previously, students who completed the required calculus course did not achieve better results in Java. We were somewhat surprised at this finding and recommend further research in this area.

As it appeared in the literature, our study found no significant difference between the performance of men and women. We have too few women entering the field. Those women that do enter, however, are as successful as men.

The fact that some of our students take the introductory programming classes as freshmen or juniors and then do not take the advanced Java class until close to graduation has been mentioned by some faculty as a problem area. This study validates this concern. Programming concepts and theory can be easily forgotten if not reinforced and applied immediately. The department may also have contributed to this problem by not offering the course every semester in the day and evening sessions. This success factor is relatively easy to implement. Students need to be advised to take the Java course immediately after completing Programming 2 and the department has to offer the course every semester, day and evening, with as many sessions as needed.

On the other hand, it may not be the delay itself that causes the later problems in the advanced programming classes. It may be that some students feel insecure with programming itself and thus delay taking the advanced course because of these feelings of insecurity. We plan further investigation in this area using follow-up student interviews.

The results of the study seem to indicate C++ may provide a better foundation for upper level programming in Java. There could, however, be any number of factors to explain this. C++ is closer in syntax to Java and may make the transition to that language easier. On the other hand, it may simply be that the better programmers tend to take C++ instead of VB. This is another area that will be well served by more research of a qualitative nature and student interviews.

It does not seem surprising that students in the programming track would do better in Java than students in the other tracks. Systems students had the worst Java grades. Systems students may have already made the decision to avoid or dislike programming. This brings up the issue of whether or not all information systems students need advanced programming. Are we forcing them to take a course they do not like and do not do well in? Will programming ever be a part of their careers? This topic requires further study outside the realm of this project.

Finally, students that have a better general grade point average do better in Java. Good study skills and habits help a student succeed in any subject. Motivational and psychological factors are important in all academic fields. Students who strive for good grades will want good grades in all their classes. Helping our students to learn and attain good study habits, organizational skills, testing practices, etc. should help students do well in Java as well as their other courses.

6. CONCLUSIONS

Based on the results of this study, students should be strongly encouraged to take Java immediately after completing Programming 2. Programming concepts and theory can be easily forgotten if not reinforced and applied immediately. The department should also do their part and offer Java in both the spring and fall semesters for day and evening sessions.

The department should consider mandating C++ as a required introductory language and offer Visual Basic as an elective. As mentioned previously, C++ is similar to Java and may make the transition to Java easier. It will also make it easier for the instructor if all students have the same background and entry level skill sets.

This study did not consider whether all information systems students need to take advanced programming. It does suggest that this matter should be researched and discussed. How many programming classes are needed for students who do not intend to become programmers?

This study was limited to only one college and this college may be different than other colleges. The results, therefore, may not be generalizable. Further research at other schools or a consortium of other schools would help to alleviate this limitation.

This study was also limited by its use of final grades as assessment measures. A student’s final grade is composed of numerous factors including class participation, objective tests, homework, etc. In this study, we were looking at only one part of this grade — success in programming. It was hard to weed out that one factor from the overall picture. In the future, we plan to give assessment tests in the programming classes to use as comparison measures.

In addition, we would like to enhance the research study by looking at some of the personal and psychological factors that may affect a student's success in the Java class. For this later study, we would like to conduct a survey and perform in-depth interviews with Java students.

7. REFERENCES

Bennedsen, J. & Caspersen, M. E. (2005). An Investigation of Potential Success Factors for an Introductory Model-Driven Programming Course. *Proceedings of the First International Workshop on Computing Education*, Seattle, WA, 155-163.

Bhatnagar, N. (2009). A Study of the Inclusion of Programming Languages in an Undergraduate Information Systems Curriculum. *Information Systems Education Journal,* 7 (84).

Byrne, P. & Lyons, G. (2001). The Effect of Student Attributes on Success in Programming. *Proceedings of ITiCSE 2001*, Canterbury, Kent, ACM Press.

Chen, T., Monge, A. & Simon, B. (2006). Relationship of Early Programming Language to Novice Generated Design. *ACM SIGCSE Bulletin*, 38(1), 495–499.

Dehinbo, J. (2006). Determining Suitable Programming Language for the Bachelor of Technology (IT) Curriculum. *Proceedings of the Information Systems Education Conference 2006*, Dallas, 23.

Guzdial, M. & Soloway, E. (2002). Teaching the Nintendo Generation to Program. *Communications of the ACM*, 45(4), 17-21.

Lahtinen, E., Ala-Mutka, K. & Jarvinen, H. (2005). A Study of the Difficulties of Novice Programmers. *ACM SIGCSE Bulletin*, 37(3), 14-18.

Lewandowski, G., Gutschow, A. McCartney, R. Sanders, K. & Shinners-Kennedy, D. (2005). What Novice Programmers Don't Know. *Proceedings of the First International Workshop on Computing Education Research,* Seattle, WA, 1–12.

Lister, R., Adams, E.S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., Mc Cartney, R., Mostrom, J.E., Sanders, K., Seppala, O., Simon, B. & Thomas, L. (2004). A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *ACM SIGCSE Bulletin*, 36(4), 119-150.

Mannila, L. & de Raadt, M. (2006). An Objective Comparison of Languages for Teaching Introductory Programming. *Proceedings of the 6th Baltic Sea Conference on Computing Education Research*, Uppsala, Sweden. 276, 32-37.

Mannila, L., Peltomaki, M. & Salakoski, T. (2006). What about a Simple Language? Analyzing the Difficulties in Learning to Program. *Computer Science Education,* 16(3), 211-228.

McCracken, M., Kolikant, Y., Almstrum, V., Laxer, G., Diaz, D., Thomas, L., Guzdial, M., Utting, I., Hagan, D. & Wilusz, T. (2001). A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-Year CS Students. Annual Joint Conference Integrating Technology into Computer Science Education. *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, Canterbury, UK, 125-180.

McIver, L. & Conway, D. (1996). Seven Deadly Sins of Introductory Programming Language Design. *Proceedings of the 1996 International Conference on Software Engineering: Education and Practice* (SE-EP '96), Dunedin, New Zealand, IEEE Computer Society, 309-316.

McKinney, D. & Denton, L.F. (2004). Houston, We have a Problem: There's a Leak in the CS1 Affective Oxygen Tank. *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Educatio*n, Norfolk, VA, ACM Press, 236-239.

Parker, K., Chao, J., Ottaway, T., & Chang, J. (2006). A Formal Language Selection Process for Introductory Programming Courses. *Journal of Information Technology Education*, 5, 133-151.

Prasad, C. & Li, X. (2004). Teaching Introductory Programming to Information Systems and Computing Majors: Is there a Difference? *Proceedings of the Sixth Conference on Australasian Computing Education*. Dunedin, New Zealand, 57, 261-267.

Rountree, N., Rountree, J. & Robins, A. (2001). Identifying the Danger Zones: Predictors of

Success and Failure in a CS1 Course. *Inroads SIGCSE Bulletin*, 34, 121-124.

Rountree, N., Rountree, J. Robins, A. & Hannah, R. (2004). Interacting Factors that Predict Success and Failure in a CS1 Course. *ACM SIGCSE Bulletin*, 36(4), 101-104.

Russell, J., Russell, B., Pollacia, L. & Tastle, W. (2009). A Study of the Programming Languages used in Information Systems and in Computer Science Curricula. *The Proceedings of the Information Systems Education Conference* 2009, Washington DC , 26.

Wiedenbeck, S. (2005). Factors Affecting the Success of Non-Majors in Learning to Program. *Proceedings of the First International Workshop on Computing Education* Research, Seattle, WA, 13-24.

Wilson, B. & Shrock, S. (2001) Contributing to Success in an Introductory Computer Science Course: A Study of Twelve Factors. *ACM SIGCSE Bulletin,* 33(1),184-188.

Systems in the Foundations of Information Systems Course to Retain Students and to Support the IS 2010 Model Curricula

Gayla Jo Slauson
gslauson@mesastate.edu

Donald Carpenter
dcarpent@mesastate.edu

Johnny Snyder
josnyder@mesastate.edu

Dept. of Business,Colorado Mesa University
Grand Junction, Colorado 81501, USA

Abstract

Systems in the Foundations of Information Systems course can be used to connect with students in computer information systems programs; a systematic approach to beginning student relationship management in this course is helpful. The authors suggest that four systems be created in the Foundations Course. These theoretical systems include an overall student relationship management system with subsystems including a personal response system, a student mentoring system, and a knowledge management system. Each proposed system is briefly described with its potential benefits, specific examples of support for the IS 2010 Model Curricula, possible costs or issues, and suggestions on how to begin.

Keywords: IS 2010 Model Curricula, Student Relationship Management (SRM), Personal Response System (PRS), Clickers, Student Mentoring, Knowledge Management Systems (KMS), Foundations of Information Systems Course

1. INTRODUCTION

Undergraduate information systems faculty who wish their programs to remain current and relevant will want to consider the IS 2010 report. As budgets decrease, it behooves every program to find ways to recruit and retain students. Following a national curricula model can be a selling point. Further, it becomes increasingly valuable for faculty in information systems programs to "exploit opportunities created by technology innovations" (Topi, Valacich, Wright, Kaiser, Nunamaker Jr, Sipior and deVreede, 2010, p. 362) while supporting program objectives and working on comprehensive student relationship management.

While many systematic methods could be pursued, four proposed systems could be particularly relevant to initiate in the Foundations of Information Systems course. This course literally provides a foundation for the other courses in the curriculum which then may be taken in a variety of sequences, and not necessarily in a particular order. Consequently, systems initiated here may be used throughout the information systems program in a variety of ways. The four suggested systems to implement in the Foundations course include an overall Student Relationship Management System (SRMS), a Personal Response System (PRS) also called clickers, a Student Mentoring System – particularly one that uses emerging technologies, and a Knowledge Management System (KMS). The latter three systems would feed data into and interface with the comprehensive student relationship management system.

The purposes of the proposed systems go beyond being part of the SRM in that they can provide specific support for various of the guidelines listed in the IS 2010 Model Curricula. These systems could also be used as part of a larger departmental effort to document retention efforts in order to obtain new or ongoing resources.

2. IS 2010 MODEL CURRICULA GUIDELINES

Revisions to the IS 2010 Model Curricula have been made through a "combined effort of numerous individuals and reflects the interests of thousands of faculty and practitioners. It is grounded in the expected requirements of industry and …is supported by …IS-related organizations." (Topi, et al., 2010, p.361) With this sort of support behind it, the IS 2010 guidelines provide valuable information for faculty endeavoring to make their Information Systems (IS) programs as relevant and up-to-to date as possible. Flexibility and agility have been built into the model when possible and a variety of programs will be able to shape their programs within it. Systems that can be used to support both the Model Curricula guidelines and additional program retention efforts are doubly valuable.

Role of the Foundations Course

Within the revised Model Curricula, the Foundations of Information Systems Course (IS 2010.1) serves to provide all students with a fundamental understanding of how information systems are designed and administered and how they benefit organizations. It is the first of the core IS courses. After taking this class, students may choose alternative sequences or electives. This class provides the initial point for building relationships with students and therefore provides the perfect place to initiate systems that may be used anywhere else in the program.

3. STUDENT RELATIONSHIP MANAGEMENT

As department and program budgets diminish, student retention becomes increasingly critical. Programs must fill classes and find resources. In some cases they must document how they are attempting to retain students. Customer Relationship Management (CRM) systems are common in business these days. Although the well-researched argument has been made that the "principles of an efficient CRM can be applied to the sector of higher education" (Hilbert, Schonbrunn and Schmode , 2007, p. 209), most Student Relationship Management Systems are not designed as comprehensively as a CRM. Effective CRM systems endeavor to track all interactions with the customer. (Kroenke, 2011) (Laudon & Laudon, 2011). SRM systems tend to be mostly focused on student admissions and registration rather than the support of academic activities within a department. (Piedade, 2008, p.1)

An SRM that is set up as an umbrella system that manages a multi-faceted strategic approach to interactions with students will contain a number of subsystems such as those described in this paper. Such a system would collect the data from these subsystems into a data warehouse and provide the analytical tools and processes to use the data effectively to enhance student relations and to document interactions. Each of the three systems initiated in the Foundations of IS Course would interface with a larger SRM.

4. PERSONAL RESPONSE MANAGEMENT SYSTEM

System Description

The first system to implement in the Foundations Course and hence the program, and perhaps the easiest to begin, is a Personal Response System (PRS) or clickers. A PRS requires wireless transmitter devices about the size of remote controls which are readily available, a receiver unit, and software. If the program's institution does not provide the receiver units, they may be obtained and carried to class by instructors. These systems allow for immediate feedback from an entire roomful of students, with each student's responses being captured and tallied. Visual feedback is also available so that graphs can be easily produced showing results of surveys or answers to questions. Students can purchase the clickers or it might be possible for the instructor to carry enough for a group. Maintenance of the clickers primarily requires occasional battery replacement.

Values and Benefits

A PRS can be useful in a variety of ways and the system can provide immediate feedback. Several schools have found PRS beneficial for student performance within their programs. (Bruff, 2008) (Mula & Kavanagh, 2009) The rates of student participation have been shown to be significantly increased. (Stowell & Nelson, 2007) One of the authors has used clickers to enhance classroom interactions by creating games that use their capabilities. They can be used to poll students and increase understanding. (Caldwell, 2007) In at least one study reported in the *Journal of Information Systems Education*, students indicated that they perceived that they participated more in classes and attended more often when the class used clickers. (Nelson & Hauck, 2008) Additionally they capture data easily for a larger SRM.

Students who become accustomed to the PRS in the Foundations course would then be comfortable using it to answer survey questions in other classes throughout the program. Furthermore, they could use their same clickers throughout their program.

Specific Support for IS 2010 Model

The PRS provides specific support for several of the Foundations of IS Course learning objectives listed in the IS 2010 Model Curricula guidelines. (Topi, et al., 2010) Here are a few examples.

First a behind-the-scenes approach may be easily used for the system to readily demonstrate information system components: hardware, software, data, and network elements. This understanding particularly supports Learning Objective 6. Students can begin to learn to recognize an information system and to "understand how…(it can)… provide the information needed to gain business intelligence…" which is part of Learning Objective 10 for the IS 2010.1 course. Additionally, as students are polled and the results displayed and adaptations to the course are made as a response to the feedback, students can be shown how the system is useful as a way to "foster stronger relationships with customers", part of Learning Objective 11. Limitations of systems and Garbage In Garbage Out issues involving data can be easily shown with a PRS by using poorly constructed questions.

Costs and Issues

Hardware and software costs to the students and faculty are quite minimal. Once purchased for a nominal price clickers can be used for as many years as necessary and afterwards could be returned, sold, given or traded to others. Perhaps the largest cost for faculty would come in terms of time spent getting questions organized. Any decent quiz or survey question, regardless of whether clickers will be used, takes time to create. PowerPoint can be used to display the questions, and the system captures the question and the answers. Sometimes systems must be set up to work on varied frequencies for each room if numerous classes are using clickers. Security is not a huge issue with a PRS as data can be aggregated anonymously and pulled into the SRM without student identifying data.

Getting Started

Check to see if the institution housing the program supports clickers by offering the software to download, receivers in the rooms, and training. Some institutions may offer prepared clicker rosters for easy setup. Some campus bookstores offer clickers for sale or rent. Publishers sometimes offer clickers as part of book packages.

If the institution does not, sites such as www.iclicker.com offer instructions on getting clickers and getting students in the class registered.

5. STUDENT MENTORING SYSTEM

System description

A student mentoring system often pairs upperclassmen with lower with the intention of building relationships that give students reasons to stay in a program. It might also be possible to create a student mentoring system that partners alumni or business mentors with students. Technology based mentoring could be part of either sort of system (Pamuk & Thompson, 2009) and social networking sites could provide mentoring tools as well. A computerized system can be used to store, pair and perhaps even evaluate applications and document mentoring interactions. A Student Mentoring System that focuses on bringing in emerging technologies can be particularly beneficial to students in the Foundations of IS Course.

One approach might be to allow students who wish to mentor or be mentored to go through an application process, and those who wish to be mentors must meet certain criteria and participate in some sort of training. Student organizations such as the student chapters of the Association of Information Technology Professionals (AITP) may provide a source for mentors, and may facilitate group events for students to interact. One of the valuable traits of a student mentor, according to the research, is being in the same program of study as the mentee. (Terrion & Lennox, 2007)

Values and Benefits

A student mentoring program, set up in a systematic way and well documented can be quite effective. (Marable, 1999); (Pisimisi & Ioannides, 2005) At least one study has shown that the experience of a student mentor provides as much value for the mentor as the mentee in retaining students. (Amaral & Vala, 2009)

Specific Support for IS 2010 Model

If designed to use emerging technologies in the mentoring process, the student mentoring system can be used to support Learning Objective 9 from the IS 2010 Model Curricula guidelines in helping students become "aware of emerging technologies that enable new forms of communication, collaboration, and partnering." (Topi, et al., 2010, p. 391) Even if emerging technologies were not a part of the student mentoring system, it could be used to support Learning Objective 11 as it assists in understanding how systems can be used to enhance relationships with customers and enforce organizational processes. Additionally, the mentoring system helps stress the need for interpersonal communication skills that are to be discussed in this course. (Topi, et al., 2010)

Costs and Issues

One issue regarding the creation and maintenance of a Student Mentoring System that pairs students up is attempting to provide screening for problematic situations. Automated screening may not be particularly effective. Beyond initial screening and matching issues, someone must be available to deal with mentor/mentee issues that arise. Perhaps less of a one-on-one approach would be preferable. In other words, have a group of mentors (perhaps the AITP club) for a group of Fundamentals students. Other issues involve protecting the privacy and providing security for the information collected on the mentoring applications.

Getting Started

The student mentoring program really needs to begin with students in the Foundations of IS course at the point when students are not completely sure that they will remain in a program. A beginning point is to create a database that tracks events that provide opportunities for mentoring. For example, have the AITP Members provide a party early in the Fall (perhaps as late as Halloween) specifically for the Foundations of IS Students as a way of initiating contacts between upperclassmen and lower while hopefully sparking interest in joining AITP early in the program.

6. KNOWLEDGE MANAGEMENT SYSTEM

System Description

A knowledge management system (KMS) can be used to enhance the sense of community and allow multiple entities to share their knowledge. (Mancilla-Amaya, Sanin & Szerbicki, 2010) The idea behind creating a KMS to use initially with Foundations of IS students would be to provide a sense of community by building a knowledge based platform, perhaps by merely using a list of categories with contact information for those considering themselves to have some expertise in given areas, and to bring in not only new students in the Foundations of IS course but also faculty, advisors, emeriti, alumni, administrators, etc. The knowledge management system could be categorized by academic and departmental or program subjects, but also by other areas of expertise. Hopefully students coming into the program would be willing to allow themselves to become a contact regarding an area in which they have expertise, particularly if the contact information contains only a student's email address on campus which is already widely available. For example a student who excels in creating animated films could be listed as a contact for those interested in that topic.

Naturally, part of the KMS could be used to provide answers and contact information regarding program requirements, course descriptions, or internships. Students who are seniors might be contacts for freshmen who are wondering about the major. Even community members who are willing to answer questions about their careers could be included. AITP members could be contacted regarding a variety of topics and events.

A wiki might be part of this KMS – providing places where questions could be answered by multiple people on an ongoing basis. Departmental oversight and maintenance of the wiki would be necessary to make it effective.

Values and Benefits

The use of the KMS in the Foundations of IS Course would be multifaceted: help them to be included in the community in some fashion, help them to see the KMS as a resource to answer their questions, and to give them a larger perspective by exposing them to the wider view of a program through alumni and community inputs. A benefit of a KMS includes "forming relationships and knowing whom to contact for help". (Santo, 2005, p.42) A student who is listed as a potential contact for an area of expertise, be it repairing mountain bikes or writing Excel macros, will feel a part of a community in a way that a student without that listing cannot. The KMS can be used to share internal and external knowledge.

The KMS can also benefit the department: "The purpose of having knowledge-based systems is to be able to make more informed, research-driven policies and procedures that improve program and service delivery to students…" (Petrides, 2002, p. 72)

Specific Support for IS 2010 Model

The KMS as described can be used to support nearly every one of the fourteen learning objectives described in the IS 2010 Model Curricula, particularly if students are allowed to see the back end of the KMS. Especially relevant are Learning Objectives #8, #9, and #10 as they stress emerging technologies and various types of systems used for decision making. (Topi, et al., 2010)

Costs and Issues

Resources must be found to design and initiate the system. Additionally faculty teaching each of the Foundations of IS classes must find a way to get students involved with the KMS.

One potential issue for such a system might be the likelihood of some students using the KMS as a means to attempt to advertise or sell a product or service. What problems this behavior might cause must be considered prior to developing the system. If selling is not to occur, system participants must be made aware of that prohibition in advance of participating. This situation can be dealt with as it is not unlike allowing students to set up web pages under the auspices of the college web site. It may be more difficult to have community members, business owners, and alumni participating without getting into selling.

Getting Started

Perhaps an upper-division class such as Systems Analysis and Design or a Knowledge Management course such as that described in the IS 2010 Model (Topi, et al., 2010) could initiate the KMS… perhaps other program participants could become involved in managing and maintaining it.

7. CONCLUSIONS

An undergraduate information systems program that is based on the IS 2010 Model Curricula, sound educational practices, and emerging technologies will be up-to-date, relevant and provide ways to retain students. Updating the Foundations of IS Course can be particularly important as it provides the initial contacts that start relationships with program students. Initiating a Student Relationship Management System with Personal Response Systems, Student Mentoring Systems, and Knowledge Management Systems in the Foundations course and then using them throughout the program may enhance the effectiveness of the program, help support the learning objectives for the course, and assist in student retention.

8. REFERENCES

Amaral, K. & Vala, M. (2009). What Teaching Teaches: Mentoring and the Performance Gains of Mentors. *Journal of Chemical Education*, *86*(5), 630-633.

Bruff, D. (2008). Classroom Response Systems (Clickers). *Center for Teaching*, Retrieved June 28, 2010 from www.vanderbilt.edu/cft/resources/teaching_ resources

Caldwell, J. (2007). Clickers in the Large Classroom: Current Research and Best Practice Tips. *Life Sciences Education*, *6*(Spring 2007), 9-20.

Hilbert, A.; Schonbrunn, K. & Schmode, S. (2007). Student Relationship Management in Germany – Foundation and Opportunities. *Management Revue*, *18*(2), 204-219.

Kroenke,D. (2011). Using MIS, 3rd ed. Prentice Hall, NJ.

Laudon, K. & Laudon, J. (2011). Essentials of Management Information Systems, 9th ed. Prentice Hall, NJ.

Mancilla-Amaya, L.; Sanin, C. & Szerbicki, E. (2010). Smart Knowledge-Sharing Platform for E-Decisional Community. *Cybernetics & Systems, 41*(1), 17-30

Marable, T. (1999). The Role of Student Mentors in a Precollege Engineering Program. *PJE Peabody Journal of Education,* 74(2), 44.

Mula, J.M., & Kavanagh, M. (2009). Click Go the Students, Click-Click-Click: The Efficacy of a Student Response System for Engaging Students, Improving Feedback and Improving Performance. *E-Journal of Business Education & Scholarship of Teaching, 3*(1), 1-17.

Nelson, M.L. & Hauck, R.V. (2008). Clicking to Learn: A Case Study of Embedding Radio Frequency-Based Clickers in an Introductory Management Information Systems Course, *Journal of Information Systems Educaton, 19(*1), 55-64*.*

Pamuk, S. & Thompson, A. (2009). Development of a Technology Mentor Survey Instrument: Understanding Student Mentors' Benefits. *Computers & Education, 53(*1), 14-23.

Petrides, L. (2002). Organizational Learning and the Case for Knowledge Based Systems. *New Directions for Institutional Research*, 2002 (113), 69-84.

Piedade, M.B. (2008). Student Relationship Management: Concept, Practice and Technological Support, Engineering Management Conference2008, *IEEE International.* Retrieved on June 30, 2010 from http://ieeexplore.ieee.org.

Pisimisi, S. & Ioannides, M. (2005). Developing Mentoring Relationships to Support the Careers of Women in Electrical Engineering and Computer Technologies: an Analysis on Mentors' Competencies. *European Journal of Engineering Education, 30*(4), 477-486.

Santo, S. (2005). Knowledge Management: An Imperative for Schools of Education. *TechTrends: Linking Research &Practice to Improve Learning, 49*(6), 42-49.

Stowell, J.R. & Nelson, J.M. (2007). Benefits of Electronic Audience Response Systems on Student Participation, Learning and Emotion. *Teaching of Psychology, 34*(4), 253-258.

Terrion, J. & Lennox, L. (2007). A Taxonomy of the Characteristics of Student Peer Mentors in Higher Education: Findings from a Literature Review. *Mentoring & Tutoring: Partnership in Learning, 15*(2), 149-164.

Topi, H.; Valacich, J.; Wright, R.; Kaiser, K; Nunamaker, J.; Sipior, J. & deVreede, G. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Communications of the Association for Information Systems, 26(18), 359-428.

Culturally Sensitive IS Teaching: Lessons Learned to Manage Motivation Issues

Wenshin Chen
wenshinchen@hotmail.com
Department of Management, Abu Dhabi University
Abu Dhabi, 59911, United Arab Emirates

ABSTRACT

This paper seeks to raise awareness of culturally sensitive teaching that is largely overlooked in the IS teaching community. In a global, networked environment commonly faced by the contemporary business or academic world, it is imperative to prepare future IT professionals with adequate cultural understanding of such a multicultural environment in which their future work practice will engage. Derived from a teaching case situated in the context of HBCU (Historically Black Colleges and Universities) in the US, this paper narrates and reflects cultural issues and challenges that are increasingly prevalent in today's education systems. More specifically, the study analyzes motivational issues that are commonly observed in a homogeneous student group and provides practice lessons to help educators who might face similar issues in their teaching context to manage those issues. Insights gained from the study help reflect the significance of developing culturally sensitive pedagogy that might require future IS educators' and researchers' attention.

Keywords: culturally sensitive, IS education, case study, HBCU, motivation

1. INTRODUCTION

Cross-cultural teaching has not gained adequate attention in the IS community. Most empirical studies in the mainstream IS education journals tend to focus on IS teaching issues involving particular subject matters such as virtual teams , Web 2.0, and other emerging topics (Chou & Liu, 2005; Retalis & Avgeriou, 2002; Shee & Wang, 2008). In today's global, networked society and business world, multicultural groups often interact with one another. Many cross-cultural issues stemmed from group differences are embedded in most employees' routine operations (Skelton & Allen, 1999). Inadequate understanding of cross-cultural issues in the teaching and learning process would then indicate poor preparation for future IS professionals' career.

The purpose of this study thus seeks to provide empirical analysis to advocate culturally sensitive teaching practice in the IS community. More specifically, the study focuses on gaining insights of cross-cultural issues involved in the context where a homogeneous student group interacts with a diverse faculty group. Set in this backdrop, the study inquires "How faculty members can improve IS teaching and learning in a culturally challenging environment?" Broadly speaking, the meaning of culturally challenging in this study is based on the instructor's perspective and refers to significant cultural differences between the faculty group and the student group. For instance, a White male professor situated in an all Arab female class or an Asian professor situated in a Historically Black Colleges and Universities (HBCU) would be considered culturally challenging. The latter is indeed the research context where case stories narrated below occurred.

It is expected that insights gained from the study could provide several contributions to the IS community. First, it calls for attention to and raises awareness of culturally sensitive teaching practice that is critically needed in the IS world. It also helps understand a minority instructor's survival experiences in a culturally challenging environment that has not been much understood in the literature. In addition, it provides lessons to IS educators worldwide regarding how to practice culturally sensitive teaching. Lastly, it serves as an exploratory platform on which future cross-cultural studies in the IS teaching community could build.

2. CROSS-CULTURAL EDUCATION

The literature of cross-cultural education tends to focus on the multicultural significance in curriculum and instruction (Ball, Berkowitz, & Mzamane, 1998). While much discussion leads to specific educational contexts such as teacher education (Britten & Mullen, 2003), higher education (Goodwin & Nacht, 1991), and business education (Slone & Wines, 1995), a significant proportion of the cross-cultural investigation is devoted to national differences or international experiences (Fedorowicz, 1997; Smolicz, 1997). A serious concern is that little attention is paid to the teaching in general relating to multicultural and cross-cultural education. Only few researchers address the teaching implications of such cultural issues as in culturally responsive teaching (Gay, 2000), culturally centered pedagogy (Sheets, 1995), and culturally relevant pedagogy (Foster, 1995). Although slight, perhaps cosmetic, differences exist in the terms, they all emphasize the understanding and incorporation of cultural elements in teaching. Gay (2000), most specifically, defines culturally responsive teaching as:

> using the cultural knowledge, prior experiences, frames of references, and performance styles of ethnically diverse students to make learning encounters more relevant to and effective for them. It teaches to and through the strengths of theses students. It is culturally validating and affirming (p. 29).

Gay's notion is extensively related to other scholars' works (Banks & Banks, 1995). To prepare teachers for a multicultural teaching context and education, a clear understanding of Gay's scholarship will greatly facilitate the endeavor. However, there is an implicit, taken-for-granted assumption embedded in Gay's and other scholars' culturally responsive research—i.e., they all tend to assume or advocate a diverse student population and omit the teacher's identity, which is also a major shaping force. In other words, these researchers simply advocate a greater recognition of the minority students or diverse student populations (Irvine & York, 1995; King, Hollins, & Hayman, 1997) and pay little attention to a teacher's voice and background. An incorporation of a teacher's voice and identity in cross-cultural research is rather significant because it could better articulate the role a teacher plays in the teaching and learning process, most specifically with respect to culturally responsive teaching. Moreover, it could further help broaden and deepen the understanding of the realistic teaching context in the contemporary internationalized society and multicultural educational environment (Kalantzis & Cope, 1999; Schoem, Frankel, Zuniga, & Lewis, 1993)—particularly in higher education and IT education where multicultural issues are often greatly involved not just in student populations but also in teachers' backgrounds (Ball, et al., 1998). In other words, the teaching and learning process involves not just the students but also the instructors. The traditional literature of cross-cultural education that mostly focuses on cultural differences among students is thus no longer sufficient in today's internationalized education systems. The cultural differences between teachers and students (instead of merely among students themselves) should also be considered.

In the context of Historically Black Colleges and Universities (HBCU), this point can be even more articulated due to its inherited cultural differences. In the existing literature, studies of Historically Black Colleges and Universities mainly concern the general characteristics and historical context involved (Evans, Evans, & Evans, 2002; Jewell, 2002; Nettles, Wagener, Millett, & Killenbeck, 1999; Oliver, Oliver, Kolheim, & Glenn, 1996). Although it helps provide an overview of what distinguishes HBCU from other colleges and universities, cultural and technological specific aspects of college education in HBCU have almost never been addressed. For example, Oliver et al. (1996) demonstrate historical role HBCU play in the society in general and in military leadership in particular. Nettles et al. (1999) provide information regarding the geographic regions of HBCU. Evans et al. (2002) compare the enrollments of all 106 HBCU between 1990 and 1999. Jewell (2002) further notes how HBCU serves and educates African American community when the society was racially stratified.

All of these viewpoints seem to focus on the African American perspectives. Little attention is paid to teaching and learning in HBCU. For contemporary and future educators who face increasing cultural issues in everyday practice, a clear understanding of how teaching and learning evolves in a culturally diversified environment such as HBCU is essential.

3. RESEARCH METHODOLOGY

Due to its sensitive nature and unique HBCU context that could not be easily understood without real experiences lived and told, personal experience method is considered the most suitable and feasible research approach. Personal experience method has its own long intellectual tradition under the umbrella of qualitative inquiry (Denzin & Lincoln, 1994) and most suitable when the research context involves with authentic and subtle issues in a unique context that will not surface otherwise (Grant, 1999). To enable a deeper understanding of subtle issues occurred in the research context, personal experience method requires a researcher's systematic reflections on his/her participant observations and work practice (Clandinin & Connelly, 1994; Rearick & Feldman, 1999). Those reflections need to eventually be organized into convincing stories that narrate authentic issues in the research context investigated; they are thus often related to the notions of reflection in action (Schön, 1983), narrative knowing (Bruner, 1996), narrative truth (Spence, 1982), and professional knowledge landscape (Craig, 2003). In the field of education, these notions have greatly contributed to the existing body of knowledge and helped reshape teaching practice over the years. In the classroom settings, these notions argue that a teacher's knowledge is embedded in his/her teaching practice and thus could not be better understood than his/her own professional reflection (Clandinin & Connelly, 1988). It is thus not surprising to observe that personal experience method is widely used in educational settings (Clandinin & Connelly, 1987) and cultural studies (Bruner, 1996).

The HBCU teaching experiences narrated in this study occurred during one academic year when I taught a Java Programming course two sessions a week. At the end of each teaching day I reflected my teaching experiences in a personal journal. In addition to such autobiographic reflections, my understanding of the teaching (and research) context was also derived from a great deal of interaction with my colleagues' experiences in the professional setting that created a professional knowledge landscape (Craig, 2003). During that academic year, email messages occurred in the teaching context were also saved and later retrieved and analyzed. More than five hundred received email messages helped build a more holistic understanding of my lived experience during that academic year.

4. CASE STORIES

4.1 Organizational Context

The institution where my teaching took place was a public university located in southwest U.S.A. When I first interviewed for the position, I, an Asian Ph.D. student at that time, was not aware of its history. During the interview, the Dean questioned, "We have lots of underprivileged kids here. How would you motivate them?" As I walked around the campus after the interview, I realized that almost all the students were African Americans. The link between African American, on the one hand, and economically underserved students, on the other hand, began to emerge as major themes characterizing the population of the institution of which I would soon be a part. Prior to teaching in this institution, I taught various courses for four years in two different public universities. Two years earlier, I also taught the same course and obtained a high teaching rating in another university nearby where student population had one of the highest diversity indexes in the nation.

My primary contacts in the college included departmental colleagues: Dylan, an assistant professor from Canada who was recruited at the same time as me, Mooney, an Indian assistant professor who was hired a year earlier, Dr. Om, an older assistant professor who came from Africa, Sharon, a full time adjunct professor who was a White American born and raised in the State, and my officemate, Ray, also a full time adjunct professor who was an African American. To protect confidentiality and privacy, I used pseudonyms for all characters appeared in the study. My interaction with these colleagues, among others, helped me better reflect my personal teaching experiences and broaden my narrative understanding of these teaching stories.

4.2 The First Encounter

When I first stepped into the classroom, not a hint of any expression was on anyone's face. Before my first lecture began, the following conversation occurred.

> Me: "Have you taken any programming course before?"
> Students: "Yes!"
> Me: "What kinds of programming languages?"
> While someone said "Structured programming" others shouted "Visual Basic!"
> Me: "What did you learn in Visual Basic class?"
> Students: "Nothing!"
> Then they looked at one another and laughed as if that course was a joke.

As I came to realize that I might have to build their understanding of programming from the basics Sharon's advice for me to "switch gears" quickly emerged.

> You have to *switch your gears*. Students are different here. Do you know what is the average of SAT scores here? 650…They don't have the motivation to do what it takes.

4.3 Stories of Motivation

In addition, students' attitude toward the textbooks also left a striking impression on me. Most students did not even have a textbook two weeks after the semester started. One faculty member even emailed to the entire faculty and indicated that the situation was less than tolerable. His email might have resulted from his sheer frustration and cultural shock that we all collectedly experienced. Two kinds of frustrations typically resulted from the issues of student's motivations: (1) some of them demonstrated negative motivation toward the course, and (2) others disorganized priority at school. To avoid repetition, only Mitch and Kalere's stories were narrated below to show the first type of issue and Neil's and Ella's stories for the second type.

4.3.1 Negative motivation

Mitch was a student in the first semester. The first few weeks when the class was conducted in a regular classroom, Mitch would come in and sit by the window looking out to business school's parking lot. He did not seem to be one of those members who were close to any social group and thus always appeared in the hallway or classroom alone. When later the lectures were conducted in a computer laboratory, he would choose to sit in one of the last rows and hide himself behind a computer. Although I often attempted to attract students' attention by interacting with them during the lectures, he mostly could not answer my questions.

To encourage him to visit me for further assistance, I reminded Mitch at every opportunity that presented itself. A student such as Mitch who showed little interest for the class and demonstrated little progress throughout the process usually would not pass the course let alone achieve high marks. Mitch eventually came to visit me occasionally. The questions he asked were usually very basic and were expected to be understood in previous chapters. It was common that I explained details of beginning concepts from the first few chapters. A polite student as him might merely perceive that visiting me for assistance was for me because I "requested." The initiative did not stem from his intention of learning but rather from fulfilling my 'requirements' for the class. Nevertheless, he fulfilled basic requirements and passed the course with an average score. He would later graduate on time and come back to request a letter from me for his MBA application.

In contrast to Mitch's disinterest, Kalere's motivational issues appeared to involve different matters. Since the first day of second semester, Kalere had never showed interest in being my class. When the class was conducted in a regular classroom, she would come to the class with a stern face and sat through it without a word. She and her friend would always sit together and murmur among themselves as the lecture proceeded. Once when I conducted a lecture in a computer laboratory, Kalere apparently lost interest in my lecture and started combing her hair instead. As she started slowly combing her hair in front of the entire class, my lecture was almost disrupted because I could not believe what I just observed.

Her low motivation was also shown in the first assignment submitted. Kalere and her close friend Venetia, among very few others, only submitted hand written assignments. Students such as Kalere often detached themselves from the instructor and the course; communication and interaction was reduced to a minimum. The first time we had a conversation was when she quickly came and announced that she would not be in the class because she needed to deal with student organization affairs downstairs. As I passed through downstairs hallway after the class, Kalere and her fellows of student organization were making macaroni cheese on a little school table and handing plates to students who passed through. In Kalere's case, what I could do was to remind her that she needed to at least fulfill all necessary requirements so she would not have to come back again. When the final grade was calculated, it was probably a great relief for both of us that she indeed achieved just that objective.

4.3.2 Disorganized priority

In contrast, Neil and Ella's motivational issues stemmed from a different type of source—the disorganization of their priorities at school. Neil was a noticeable student because he always sat right in front of the classroom podium. He sometimes came to the classroom, paid attention and responded to the questions; other times, he might just fall into his sleeping zone. His attendance was significantly lower than most students'. As I once entered the classroom, Neil was requesting one of students to take some notes for him and immediately left the classroom without acknowledging my presence not to mention explaining his absence.

Neil's problem, largely struggling between school work and extracurricular activities, finally began to manifest itself when he came to speak with me about considering dropping the course. After understanding his concerns, I explored his options between dropping the course immediately and trying to continue and pass the course. As he revealed that his main objective was merely to pass the course and move on, I encouraged him to continue because the possibility for him to pass the course was still likeable. In addition, in the worst scenario, even if he failed eventfully, he would have learned a general picture of the course and thus built a better foundation for his next trial. He seemed to take my reasoning rather well and decided to continue. Three days later, Neil showed up at my office with an add/drop form in his hand. He explained that he reconsidered it over the weekend and leaned toward staying in the class if there was still reasonable hope for him to pass the class. I reviewed the record once more and explained to him that the chance for him to pass the course was manageable. After my explanation, he then decided to continue but also requested special tutoring. Nevertheless, when the lecture started that day, Neil's eyes were only fixed on the floor. When the first computer programming exam was held in a computer laboratory, the only person who did not attend was Neil. Surprisingly, his pictures soon appeared everywhere on campus—"Vote for Neil Richmond." Now the specific reason that he was distracted was revealed—he was running for the president of the student government. To him, that took priority over his college education.

Later, I met him by chance in a hallway and wondered why he never officially dropped the course. He revealed he has asked another professor's signature to do so already. Interestingly, Neil reenrolled in the same class the next year. To his surprise, the course was still instructed by me. To my surprise, he was much more focused the second time and managed the course better. My observation led me to believe that he then understood certain intellectual thinking was required to pass this technical course. Although after he turned in his midterm exam, he exhaled to me "oh, I hate thinking," he received a decent grade the second time without too much struggle.

Similarly, Ella did not seem to prioritize her life and schools effectively. Her goal had always been to only pass the course. In a class where most students detached themselves from the instructor, it was not difficult to notice a quiet and friendly student such as Ella who would often come and visit for further assistance. In the beginning, I was delighted to assist her but soon realized that she was concerned less about understanding the materials than about building good relationship with me. For instance, her first visit was not about classroom materials but about a request for a copy of CD from the book. Apparently, she did not intend to purchase the textbook but hope to make a copy of CD from my book. As our conversation continued, her family and work conditions were soon revealed. A mother of a little boy and with a boy friend and a job in the National Guard on campus, she expected to graduate the following year and to maintain the same job. Before the second exam in which a paper section and a computer section were expected, she visited me again and we had an interesting conversation.

> Ella: "Are you gonna really have the computerized exam?"
> I was stunned because the format has been publicly announced during the lecture repeatedly. As such, my reaction was: "Yes, why do you even need to ask?"
> Ella: "Well, I was hoping you would say no."

Ella's first reaction when she handed me her computer exam was, "This is hard!" Similar situation occurred before the final exam. Ella visited my office a few days before the final exam. Her intention, again, was less about preparing for the final exam than about 'softening' my standard for her. In addition to her family condition, she described her needs to pass the course because of the requirement of her military financial aid. Her request was eventually explicit and direct—"Please give me a C!" Before she walked out of final exam room, she turned and whispered to me, "Give me a C!" Her final score turned out to be the second lowest in the class. After receiving her grade, she emailed me to tell her disappointment. I could simply line out her grade profile in a dim hope that she would eventually understand that her grade was determined by her own performance not by her 'talk' with me. Ella registered in the next Fall semester. Perhaps she has learned from her first experience and became more dedicated to the course in a professional way, and eventually advanced her graduation plan on schedule.

4.4 The Reshaping Process

The issue of students' motivation first occurred to me when the Dean interviewed me. While the frustration was overwhelming in such teaching contexts, it was also an opportunity for me to develop my understanding of the students' social and cultural backgrounds and, in turn, reshape my teaching approach as my narrative knowing evolved. Several subtle changes were gradually incorporated into my teaching. First, the way I addressed the requirements of the course including textbook, computer software, exams, and assignments has changed to accommodate the students' needs. While many instructors were frustrated with students' disinterest in purchasing textbooks, I reexamined my goal of teaching and sought ways to encourage their learning even without required materials. Although the course was technical and difficult by nature, I emphasized that my focus would be on their overall effort and understanding instead of simply the test results. Since I was also a student at that time, I connected to their issues concerning the burdensome expenses of higher education in the US. If a student could not afford to purchase textbooks, I understood their desperation but encouraged them to seek other sources from classmates or me.

The use of learning management tools further allowed me to post lecture materials so that the impact of not having a book could be minimized. When some students complained that they did not have a computer or the Internet at home, I simply related to my own experience as I also lacked of the Internet and explained how to utilize local library's computing resources to access online learning materials or submit assignments or projects. The fact that I worked two part time jobs and continued my doctoral study at the same time also helped me to better sympathize with the difficulty they were facing in raising families and continuing college education at the same time. My intention was to help them recognize that there might be more efficient ways of managing their life and school simultaneously. Better utilization of technology and relevant resources and better organization of their priorities were two of those examples.

5. REFLECTIVE LESSONS

In reflection, several lessons could be derived from stories narrated above. First, a clear understanding of cultural context of student population is imperative to enable an effective teaching approach. In Mitch's case, he was a shy and polite student who clearly was disinterested in the course; Kalere valued social relations more than classes; Neil prioritized extracurricular interests over course requirement; Ella focused on networking with professors instead of understanding materials. All of these cases required the instructor to find ways to better manage motivation issues. In Mitch's case, I requested his attention and even official visits for extra tutoring; in Kalere's case, the bottom line was to make her realize that the best interest for her was not to come back again; in Neil's case, my approach was to be specific about passing possibility and the effort required and allow him to reason with himself—it was too late for the first time but served the purpose the second time; in Ella's case, my message was very firm about class integrity in which students were expected to earn their grades with their performance not with their networking—it also eventually accomplished the objectives in the second trial.

One of approaches that could shorten the differences between a non-African American instructor and students in a HBCU was to relate the students' situation to the instructor's personal experiences. I found this approach particularly effective in managing students' lack of motivation. For instance, the majority of students struggled between jobs, family, and college education. As working two jobs with long commuting and completing a doctoral degree simultaneously, I explained my own situation and shared my time and emotion management experiences. Once the students realized that their problems were not unique to themselves, they became more receptive to the instructor's advice and in turn sought alternatives in life or at school that they might not previously experience.

Another important lesson learned was that it would be better for the instructor to be flexible in the teaching and learning process. When a large proportion of students struggled between their jobs and college commuting, a more flexible teaching approach would be much appreciated. For instance, online learning tools helped provide materials accessible anywhere. Textbooks that were rarely purchased by students should not be the sole source of teaching information. The fundamental requirement was to provide the same teaching standard so the students could receive similar training as those in other institutions.

6. IMPLICATIONS

Maybe the situation faced in HBCU could be attributed to its historical backdrop where non-African American professors would normally face difficulty to connect to the homogeneous African American students and in turn shapes a teaching context that is particularly challenging from the outset. However, cultural differences faced by educators in today's global, networked teaching environment do not necessarily occur in the HBCU context only. For educators, how to resolve or shorten such cultural differences will provide valuable implications in the IS teaching and learning process in contemporary higher education systems and in HBCU in particular. A general approach as shown in my teaching case is to be more culturally sensitive. In other words, the instructor will need to first understand the inevitable differences but at the same time connect similar personal experiences to the students' real life problems. This approach certainly requires more compassion and dedication from the instructor. In reality, this could mean that the instructors will face greater difficulty and challenge that might be beyond their expected work routine.

For example, when an instructor faces a large group of students, it would become nearly infeasible to deal with each individual student's social and cultural issues. This situation, as commented during ISECON (Information Systems Educators Conference) presentation, could provide an opportunity for instructors to convey their culturally sensitive mentality to their students in a different way. What students in this situation would need to know is that their instructor is sensitive to their needs and will be there for them when requested, and that in reality their instructor, just like them, faces inevitable social, cultural, and political difficulties that require their understanding and collaboration to enable a mutually respectful sensitive teaching and learning environment. Nonetheless, this proposition is beyond the scope of this paper and would require future research endeavor to further investigate.

For researchers, the implications derived from the teaching case suggest that many potential topics in relation to culturally sensitive IS teaching are emerging. First, little is known in the literature about motivation issues in HBCU, particularly from a cross-cultural perspective. There is even less information about a non-African American instructor facing a homogenous black student group. For instance, will teaching pedagogy vary when the same minority instructor faces a different student population, namely all Arab students? Or where faculty group is homogeneous and student population is heterogeneous, such as an African American female instructor in a department with all White male professors?

7. CONCLUDING REMARKS

While these insights and implications provide practical lessons for teaching practice and research, this study is inevitably limited by several empirical issues. First, it is situated in a single teaching context. To draw broader implications, studies in other contexts involving gender, racial, national and/or other differences might be necessary. Second, personal experience method chosen might raise concerns that the interpretation might vary if done by researchers with different backgrounds. Fortunately, the research context consists of a diverse faculty group where much interaction among them also helps provide deeper reflection in the case stories. However, future studies could explore these issues in other teaching contexts where different interaction between instructors and students occurs.

Despite these limitations, the study's authentic experiences help provide insights as to how an instructor's cultural awareness might help achieve the students' learning objectives. As such, the purpose of this study to draw attention to culturally sensitive teaching is served. In today's diverse higher education systems, a better understanding and higher awareness of cultural context in which the teaching and learning process is situated should help motivate students, improve student services, and ultimately enhance IS education. Future research could thus build on these insights to broader IS teaching contexts and help build more culturally connected teaching and learning systems globally.

8. REFERENCES

Ball, H., Berkowitz, S. D., & Mzamane, M. (Eds.). (1998). *Multicultural education in colleges and universities*. Mahwah, NJ: Lawrence Erlbaum Associates.

Banks, J. A., & Banks, C. A. M. (Eds.). (1995). *Handbook of research on multicultural education*. New York, NY: Macmillan Publishing.

Britten, J. S., & Mullen, L. J. (2003). Interdisciplinary digital portfolio assessment: Creating tools fro teacher education. *Journal of information technology education, 2*, 41-50.

Bruner, J. (1996). *The Culture of Education*. Cambridge, MA: Harvard University Press.

Chou, S.-W., & Liu, C.-H. (2005). Learning effectiveness in a Web-based virtual learning environment: a learner control perspective. *Journal of Computer Assisted Learning, 21*(1), 65-76.

Clandinin, D. J., & Connelly, F. M. (1987). Teachers' personal knowledge: What counts as personal in studies of the personal. *Journal of Curriculum Studies, 19*(6), 487-500.

Clandinin, D. J., & Connelly, F. M. (1988). Studying Teachers' Knowledge of Classrooms: Collaborative Research, Ethics, and the Negotiation of Narrative. *The Journal of Educational Thought, 22*(2A), 269-282.

Clandinin, D. J., & Connelly, F. M. (1994). Personal experience methods. In N. K. Denzin; & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 413-427). London: Sage Publications.

Craig, C. (2003). *Narrative Inquiries of School Reform: Storied Lives, Storied Landscapes, Storied Metaphors*. Greenwich, CT: Information Age Publishing.

Denzin, N. K., & Lincoln, Y. S. (Eds.). (1994). *Handbook of qualitative research*. London: Sage Publications.

Evans, A. L., Evans, V., & Evans, A. M. (2002). Historically Black Colleges and Universities (HBCUS). *Educational leadership, 123*(1), 3-16, 180.

Fedorowicz, H. M. (1997). The multicultural society in central Europe: Towards a communication strategy. In R. J. Watts; & J. J. Smolicz (Eds.), *Cross cultural communication: Cultural democracy and ethnic pluralism* (pp. 77-92). Frankfurt, Germany: Peter Lang.

Foster, M. (1995). African American teachers and culturally relevant pedagogy. In J. A. Banks; & C. A. M. Banks (Eds.), *Handbook of research on multicultural education* (pp. 570-581). New York, NY: Macmillan Publishing.

Gay, G. (2000). *Culturally Responsive Teaching: Theory, Research, and Practice*. New York: Teachers College Press.

Goodwin, C. D., & Nacht, M. (1991). *Missing the boat: The failure to internationalize American higher education*. Cambridge, UK: Cambridge University Press.

Grant, C. A. (1999). *Multicultural research: A reflective engagement with race, class, gender and sexual orientation*. London: Falmer Press.

Irvine, J. J., & York, D. E. (1995). Learning styles and culturally diverse students: A literature review. In J. A. Banks; & C. A. M. Banks (Eds.), *Handbook of research on multicultural education* (pp. 484-497). New York, NY: Macmillan Publishing.

Jewell, J. O. (2002). To set an example: The tradition of diversity at historically black colleges and universities. *Urban education, 37*(1), 7-21.

Kalantzis, M., & Cope, B. (1999). Multicultural education: Transforming the mainstream. In S. May (Ed.), *Critical multiculturalism: Rethinking multicultural and antiracist education* (pp. 245-276). London: Falmer Press.

King, J. E., Hollins, E. R., & Hayman, W. C. (Eds.). (1997). *Preparing Teachers for Cultural Diversity*. New York: Teachers College Press.

Nettles, M. T., Wagener, U., Millett, C. M., & Killenbeck, A. M. (1999). Student retention and progression: A special challenge for private historically black colleges and universities. *New direction for higher education, 108*, 51-67.

Oliver, T. G., Oliver, P. R., Kolheim, C. M., & Glenn, L. (1996). Historical black colleges and universities: Civilian and military leadership. *Educational leadership, 117*(2), 202-212.

Peshkin, A. (1988). In search of subjectivity-One's own. *Educational research*, 17-21.

Rearick, M. L., & Feldman, A. (1999). Orientations, purposes and reflection: A framework for understanding action research. *Teaching and Teacher Education, 15*(4), 333-349.

Retalis, S., & Avgeriou, P. (2002). Modeling web-based instructional systems. *Journal of information technology education, 1*(1), 25-41.

Schoem, D., Frankel, L., Zuniga, X., & Lewis, E. A. (Eds.). (1993). *Multicultural teaching in the university*. Westport, CT: Praeger.

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York: Basic Books.

Shee, D. Y., & Wang, Y.-S. (2008). Multi-criteria evaluation of the web-based e-learning system: A methodology based on learner satisfaction and its applications. *Computers & Education, 50*(3), 894-905.

Sheets, R. H. (1995). From remedial to gifted: Effects of culturally centered pedagogy. *Theory Into Practice, 34*(3), 186-193.

Skelton, T., & Allen, T. (1999). *Culture and global change*. London: Routledge.

Slone, R. R., & Wines, R. T. (1995). Management education reform: Opportunities for metropolitan business schools. In D. M. Johnson; & D. A. Bell (Eds.), *Metropolitan universities: An emerging model in American higher education* (pp. 273-285). Denton, TX: University of North Texas Press.

Smolicz, J. (1997). In search of a multicultural nation: The case of Australia from an international perspective. In R. J. Watts; & J. J. Smolicz (Eds.), *Cross cultural communication: Cultural democracy and ethnic pluralism* (pp. 51-76). Frankfurt, Germany: Peter Lang.

Spence, D. P. (1982). *Narrative Truth and Historical Truth: Meaning and Interpretation in Psychoanalysis*. New York: Norton & Company, Inc.

Establishing and Applying Criteria for Evaluating the Ease of Use of Dynamic Platforms for Teaching Web Application Development

Johnson Dehinbo
Dehinbooj@tut.ac.za
Department of Web & Multimedia Computing,
Tshwane University of Technology, Pretoria, South Africa

Abstract

The widespread use of the Internet and the World Wide Web led to the availability of many platforms for developing dynamic Web application and the problem of choosing the most appropriate platform that will be easy to use for undergraduate students of web applications development in tertiary institutions. Students beginning to learn web application development would not perform at their best capacity level if the platform chosen by their institution is difficult to use. This is important given the recurrent write-compile-test-recompile efforts that take place in a timed students' practice session. As part of the framework for determining the most suitable platform(s) for teaching dynamic web applications development in tertiary institutions, this study adopted an interpretive research approach to establish a set of criteria from theory and practice for evaluating the ease of use of the platforms. These criteria were tested by evaluating four platforms namely Java Servlets, Java Server Pages (JSP), Active Server Pages (ASP) and PHP using various research methods including descriptive inquiry, document analysis and observations. While Java Servlet was found to be most suitable on applying the criteria, the significance of the study lies in the establishment of a comprehensive but specific set of criteria that can be used as a scientific basis for selection.

Keywords: Web application development, programming, languages, platforms, usability, ease of use

1. INTRODUCTION

This study established part of a framework containing various criteria that can be used to evaluate dynamic Web application development platforms in terms of their suitability for teaching Web applications development in tertiary institutions. This part of the framework determines how to choose the platform that would be easiest to use by undergraduate students. The aim of the study was not to persuade readers that one platform is better than another, but to help readers make an informed decision on the ease use of the platforms.

There are currently many platforms for developing dynamic application (that are interactive, back-end database enabled, session-enabled, and with processing capabilities) on the World Wide Web. According to Lim (2002) information systems/ computer science departments need to reexamine their curricula in order to prepare students to face the challenge of being productive in a computing world swamped with web technologies. We believe that the productivity of the students would be enhanced by the choice of a platform that is easy to use in their web application development efforts. The choice made may affect the speed and efficiency with which students can develop the web applications. Web application development students would not perform at their best capacity with a platform that is difficult to utilize.

A good way of teaching Web application development would be to teach using a platform that is suitable for the students. The problem however is how best to make a reasonable choice of platform from the various platforms. It is assumed that this would involve analysis and evaluation in terms of various factors in line with the idea put forward by Ashenfelter in the statement below:

> Web development tools need to be analyzed in terms of its purpose (what it is designed to do), technology (ease of use, robustness, scalability, security, performance, etc.), support (portability, cost, ISP support), and how well it works in the real world. (Ashenfelter, 1999:109)

Thus, by implication, ease of use is one of the critical factors to be taken into consideration in choosing the platform to be used. The research question then is: How could one scientifically determine the dynamic web application development platform that will be easy to use for undergraduate students in tertiary institutions? This is in line with Ashenfelter's (1999:105) assertion that before analyzing tools, it is worth discussing how to evaluate them.

The objectives of the study are therefore:

1. To establish criteria to investigate the ease of use of dynamic web application development platforms;
2. To apply the established criteria to evaluate specific web application development platforms.

The main benefit of the study will be the educative analysis and evaluation in choosing suitable dynamic web application development platform to enhance the performance of learners, thereby increasing their potentials in such a way that would lead to higher productivity for the students.

The rest of the paper is structured as follows. Section 2 discusses the literature review highlighting the uniqueness of the study. Section 3 presents the research approach and methodology. The criteria were established in section 4 and applied in section 5. Section 6 concludes the study.

2. LITERATURE REVIEW

Analyzing, evaluating and comparing programming languages, development platforms and tools are important but also difficult as illustrated in the following statement:

> Comparisons across programming styles, or paradigms, are difficult to carry out, but are nevertheless important for understanding how different styles of programming affect the learning of novice programmers. (Wiedenbeck et al., 1999:105)

Apte *et al.* (2003) note that a study of existing literature showed varying conclusions about the superiority of one dynamic Web platform over another. Prechelt (2000) indicates that when it comes to the advantages and disadvantages of various programming languages and development platforms, programmers and computer scientists alike usually hold strong and different opinions. This is evident in the different approaches to the comparisons and selections of programming languages and platforms in the literature ranging from those based simply on intuition to those based on comprehensive criteria.

However, most of these studies did not use any explicit criterion as a basis for their comparisons, but are based on intuition rather than scientific facts. This study however believes that scientific evidence is required to support the various assertions. Comparison should be based on a variety of factors supported by scientific facts relevant literatures, experience and empirical observation results. This would involve using specific factors or criteria in the comparisons as in the next section.

Towards incorporating various criteria in their comparisons

Vinoski (2003) realizes that various comparisons of programming languages and platforms concentrate on performance comparisons. These include Renaud *et al* (2003), Cooper (2001) as well as Marshak and Levy (2003). A suitable framework for comparison should involve other relevant factors. Vinoski (2003) therefore shows that, depending on the application, other qualities such as scalability, flexibility and adaptability, ease of use, tool support, and standards conformance could very well take precedence over performance.

This study therefore proposes that the advantages and strengths of each platform should be examined in the light of certain desired qualities relevant to the criteria of interest to a specific use. For example, the ease of use of a platform would increase productivity in institutions training middle-level workers than the flexibility of the platform. It is on this note that it is deemed necessary to focus on criteria of interest towards enhancing the ease of use of the platforms by undergraduates. This leads to the review various studies on usability as presented next.

Towards a framework for usability estimation of web platforms

The ease of use of platforms for teaching web applications development is closely related to the concept of usability, except that here the focus is on knowing how usable the tools used to design the software products are, rather than the usability of a designed application software product. According to Preece *et al.* (2002) cited in Singh and Kotze (2002), usability is generally regarded as ensuring that interactive applications are easy to learn, effective to use, enjoyable, and involve the optimization of user interaction with these products.

Ashenfelter (1999:111) notes that Web databases require a knowledge that spans the worlds of Webmaster, database administrator, and programmer. Therefore, learning to use dynamic Web platforms is relatively difficult compared to other programming languages. However, the ease with which new tools can be learned has an impact on how much they will be used. Ashenfelter (1999:112) identifies a number of factors that can affect how easily a Web platform can be learned. Firstly, unnecessarily complex platforms are the biggest obstacle to ease of learning and thereby affecting ease of use. Secondly, familiarity of the related components of the dynamic Web platform will enhance learning. Thirdly, availability of useful documentation will be of great assistance in learning and using the platform.

Singh *et al.* (2001) states that several researchers have over the years produced sets of principles or guidelines aimed at improving the usability of interactive systems. These as listed in Singh and Kotze (2002) include aspects such as effectiveness, efficiency, safety, utility, "*learnability"*, flexibility, robustness, "*memorability"*, *etc*.

Adeyemo (2000) states that on the use of applications, it really looks like "Usability or else!" for the business. Although Adeyemo focused on ease of navigation of websites, some possible usability measurement criteria adapted from Keller (1990:287) were listed, which include percentage of task completed per unit time (speed metric); ratio of successes to failures; time spent in errors; percentage of competitors that do this better than current product; number of commands/statements used; frequency of help or documentation use; time spent using help or documentation; percentage of favorable or unfavorable comments; number of good and bad features recalled by user; number of available commands not invoked; number of times the user need to work around a problem; number of times the user expresses frustration or satisfaction etc.

Wesson and Van Greunen (2002) summarized usability data as comprising the three aspects of the definition of usability: efficiency, effectiveness and user satisfaction: Effectiveness relates the goals of using the product to the accuracy and completeness with which these goals can be achieved. Common measures of effectiveness include the percentage task completion, frequency of errors, frequency of assists and frequency of access to help or documentation (Wesson & Van Greunen, 2002); Efficiency relates to the level of effectiveness achieved to the quantity of resources expended. Efficiency or performance is generally assessed by the mean time taken to achieve the task; Satisfaction or acceptance describes a user's subjective level of satisfaction when using the product (Wesson & Van Greunen, 2002).

Dix *et al*. (1998) put forward principles to support usability in three categories: "Learnability", flexibility and robustness. "Learnability" refers to the ease with which new users can begin effective interaction and then attain a maximal level of performance. Usability principles related to learnability include predictability, synthesizability, familiarity, generalizability and consistency. Flexibility refers to the multiplicity of ways the user and the system exchange information. Usability principles related to flexibility include dialogue initiative, multi-threading, task "migratability", "substitutivity", and "customizability". Robustness refers to the level of support provided to the user in determining successful achievement and assessment of goals. Usability principles related to robustness include 'observability', recoverability, responsiveness and task conformance.

However, for dynamic web application systems, the focus of usability is on the features of the programming language or web platform being used in the development of the system. Thus, in summary, we could think of evaluating usability as necessary for the following reasons adapted from (Adeyemo,2000): educating other computer professionals; increasing satisfaction; decreasing anxiety (which server should I use, which database? etc); increasing trust of technology; increasing consistency across products; decreasing learning time- how long does it take for typical designers to learn to use the platform; increasing performance - how long does it take to perform tasks and how many and what kinds of errors are commonly made during typical use; increasing retention over time; increasing productivity- how many source line of codes (SLC) can be written per man-month and how easy to understand error messages and how long does it take to work around the error; establishing scalability - adaptability to growth in size of a system e.g. increased users or large databases; good development environment and validation tools; availability of add-ons e.g. email; support for software reuse and maintenance; security of information *etc*.

These metrics provide a measure of usability against which the platforms can be tested. However, since no single platform will likely positively satisfy all the criteria, there is the need for a systematic way to determine which platform will satisfy a greater number of the vital criteria. This leads us to the research design and methodology presented in the next section.

3. RESEARCH DESIGN AND METHODOLOGY

An interpretive research approach is used involving elements of descriptive, analytical and comparative studies. McMillan and Schumacher (2001: 33) state that while a descriptive study describes a system with the aim of characterizing it as it is, by using numbers, comparative study investigates the differences, thereby taking descriptive study a step further. A descriptive approach is used to characterize desirable features of the platforms towards establishing a set of criteria. In applying the established criteria, an analytical approach is used to analyze and evaluate platforms to determine the level of satisfaction of the criteria in order to compare the platforms with the goal of determining their ease of use towards choosing the most suitable platform.

Research methods adopted to obtain the results

The descriptive method used involves document review and study of various manuals and textbooks for different platforms from various sources and established body of knowledge to identify features that could enhance the ease of use of platforms for web application development. These features were characterized to establish criteria for evaluating the ease of use of platforms by undergraduate students. The established criteria serve as a model. Bowling (2002:141) describes models as abstract representations of the essential characteristics of phenomena of interest that make explicit the relationships and comparison between the characteristics. With the model in place, the stage is set for application of the criteria by analyzing, evaluating and comparing specific platforms.

The analytical method is aimed at evaluating specific platforms to ascertain their level of satisfaction of the established criteria. Documents were reviewed for the specific platforms to be evaluated and compared using various sources and established bodies of knowledge. Moreover, there is the need to physically examine the ease of use of the dynamic web application development platforms. Bowling (2002:358) describes observation as a research method in which the investigator systematically watches, listens to and records the phenomenon of interest. In this study, the use of observation was limited to the inquiry into the features and facilities available on the installed platforms and their corresponding Integrated Development Environments (IDEs) as well as on observations from experience on the use of features of the platforms for web application development.

Answers were sought to the questions and the availability of features that serve as the criteria from established sources which include those written by the designers and originators of the platforms. These were augmented by authoritative websites including those of Sun Microsystems and Microsoft as well as PHP websites and various websites for the applicable web servers such as Internet Information Service (IIS) and tomcat. Scores were assigned to each platform based on the availability of desired features, the level of support enabled on specific tasks or the inherent characteristic of the platform such as readability of program codes.

Measuring scale used

Using close-ended "Yes/No" questions, the measuring tool has values on a scale of 1 to 3, where:

3 = "Yes",
2 = "Not quite or with some workaround", and
1= "No".

A scale of 1 to 3 is used to avoid subjective situations where it will be difficult to distinguish between, for example, a score of 3 or 4 in a scale of 1 to 5. The use of the 1 to 3 scale therefore reduces the situation as to whether or not the facility is available, or in between.

Establishing reliability and validity

A measuring scale or instrument should be consistent and reliable. It should produce more or less the same accurate results every time it is applied, even when applied by different persons (Coertze & Heath, 1997: 78). Coertze and Heath (1997: 79) indicate that validity is concerned with soundness or the effectiveness of the measuring instrument. As a way of increasing validity, answers to the criteria questions were sought from recognized textbooks, authoritative websites, and journal articles. We provided accompanying references so that interested readers can verify or seek more information.

Also, to increase reliability, the quantitative characterization and evaluation using numbers would enhance clarity in the choice of platform with the highest score. This is unlike just using qualitative sentences to evaluate the platforms, at the end of which it would be difficult to say which platform is really more suitable. Furthermore, the avoidance of subjectivity by the range of values is aimed at enhancing reliability.

Data analysis

Qualitative analysis was used in the establishment of the criteria as well as in the characterization involved in the application of the criteria. Simple parametric analysis involving means was later used in selecting the most suitable platform. The scores for all the criteria were summed up for each web-based dynamic platform to obtain a total score from which the platform with the highest overall score can be identified.

Limitations

A particular limitation of interpretive studies is that it gives a subjective view of the researcher. The criteria established are based on sources of knowledge currently available to the researcher and the interpretation thereof which is also based on the researcher's experience. Therefore, the study viewed as a guide. While users can add or remove certain criteria, the next section presents the contribution to the body of knowledge in form of the criteria to evaluate the ease of use of dynamic Web application development platforms.

4. THE CRITERIA DEVELOPMENT

The establishment of the criteria to evaluate the ease of use of the platforms involves identifying qualities that would ensure that the platform will be easy to use for students studying Web application development. Therefore, the criteria should determine a platform with a user-friendly IDE. In addition, the platform should be less resource-intensive and also be portable. The desired qualities that will serve as criteria to be used to evaluate the ease of the Web application development platforms is now presented.

Availability of smart Integrated Development Environment that could enhance ease of use

This involves the availability of interpreters, compilers, as well as the availability of smart integrated editing system. According to Sebesta (1996:18), the effort of training programmers and developing applications can be significantly reduced with a good programming environment. A programming environment is the collection of tools used in the development. It may consist of only a file system, a text editor, a linker and a compiler or it may include a large collection of integrated tools such as a debugger, which when encounter error, stops and switches to the editor, leaving the cursor at the point in the program where the error was detected (Sebesta, 1996:31).

It is also important whether the interface to the tool is graphical, window-based and similar to common interfaces. Bergin (1996) states the need for a platform independent graphics model and a platform independent window-button-mouse based GUI.

The following criteria are considered necessary to ensure that our desired platform has a user-friendly and smart IDE:

☑ Availability of interpreters rather than compilers; for undergraduate students, interpreters will be good for quickly executing what they have done so far, and for seeing the effects of minor changes to the programs.

☑ Availability of an IDE that is both DOS and Windows-based; this way, the system can easily be used as preferred.

☑ Availability of smart integrated editing systems that:

- ➢ Enhance maintenance by encouraging good readability of the codes, by way of automatically formatting and indenting codes.
- ➢ Are equipped with tracing and debugging aids to aid students in debugging their codes.
- ➢ Offer other useful and smart integrated editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time.
- ➢ Are equipped with context-sensitive "help options" to assist students who are 'stuck' in certain activities during their Web applications development.
- ➢ Have the ability to switch case sensitivity on and off so that it can be turned on when there is the need for many related identifiers, such that their minor differences can be represented by alternating the case of the letters of the identifiers.

5. SUMMARY OF THE RESULTS OF APPLYING THE CRITERIA

We applied the criteria to evaluate the ease of use of four platforms namely Java Servlets, JavaServer Pages (JSP), Active Server Pages (ASP) and Personal Home Page (PHP). The results are summarized below.

Availability of features of smart Integrated Development Environment

Table 1 and table 2 reflect the scoring for the dynamic Web platforms based on the criteria on the availability of features of smart IDEs. A summary of these results based on a review of various literatures, detailed observation and testing of various IDEs and from programming experience follows.

A compiler wouldn't present a result if the program is incomplete on contains an error. The use of an interpreter rather than a compiler could allow students to instantly see results of programs up to where error is encountered or up to what has been programmed so far, irrespective of whether the program is complete or not. Apart from Java Servlet in which one needs to compile the program, other platforms use interpreters such that the incremental building of the programs becomes easier for students. In the case of JSP, however, the interpretation involves internal compilation.

Also, for prompt testing and incremental building of the programs, it is better that the platform allows execution from both the DOS prompt and the browser. The Java Servlet allows execution from the DOS prompt through the use of the "Servlet runner" program (Sun Microsystems, 1997) and also from the browser. JSP and ASP are only executable via the browser. The PHP engine can be invoked as either a CGI-callable interpreter from supported UNIX platforms and Windows NT, or as an in-process Apache module that can be viewed from all HTML-capable browsers (Ross *et al.*, 2000). PHP can be used for server-side scripting via browser as well as command-line scripting via DOS prompt and Unix shell (Lerdorf & Tatroe, 2002:1).

It is important to know in which directory to put the scripts, HTML pages and graphics. The Java platforms' JCreator IDE automatically creates directories to store the project for the Java-based platforms (JCreator, 2008). The same goes for Microsoft Visual Interdev IDE for ASP (MSDN, 2008) and PhpEd for PHP (Yank, 2008). In addition, a look at all the IDEs shows that they allow menu-driven addition of Web items such as HTML pages, scripts and style sheets, which are put in the appropriate sub-directories.

Apart from creating the directory, the IDE should also show the directory tree for the project to enable the students to browse and modify files easily. This feature is available in JCreator for the Java based platforms (JCreator, 2008), and in Microsoft Visual Interdev IDE for ASP (MSDN, 2008), as well as in PhpEd for PHP (Yank, 2008).

It is important that the IDE enhances maintenance by encouraging good readability and writability by way of automatically formatting and indenting codes. JCreator as editor for Java Servlet and JSP programming automatically format and indent codes (JCreator, 2008). Similarly, Microsoft Visual InterDev, usable as editor for ASP codes, has formatting and indenting capabilities (MSDN, 2008). PHP's

PhpED editor also automatically formats and indents the codes (Yank, 2008).
Since parentheses and brackets are always in pairs, it is useful if the IDE has hotkey matching of braces, parentheses and angle brackets. Practical testing shows that the various IDEs such as JBuilder and JCreator for the Java-based platforms, as well as the Visual InterDev for ASP have this facility. PhpEd for PHP also has this facility (Yank, 2008).

Since there is usually a special format for the HTML files, scripts, *etc,* it is preferable if the IDE has customizable code formatting templates such that students can just add contents to the templates. This feature is available in JCreator (JCreator, 2008) and in Microsoft Visual Interdev IDE for ASP (MSDN, 2008) as well as in PhpEd for PHP (Yank, 2008).

An IDE that is equipped with tracing and debugging aids would assist students in debugging their codes. Examination of the various IDEs shows that they all have tracing and debugging features.

It is important that the IDE should offer other useful and smart editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time. Examination of the various IDEs shows that none have this feature.

Given that colors enhance visualization, it is of some importance that the IDE provides syntax highlighting with configurable colors. Practical testing of the various platforms' IDEs shows that they all have this feature.

Another desirable feature is the use of functions and variable name auto-completion which would eliminate the use of wrong names and also speed up the coding process. The IDEs for the Java-based platforms have this feature. Unlike the IDE for ASP, PhpEd for PHP also has this feature (Yank, 2008).

Since the IDE knows the functions applicable in a platform, it will be useful if it has pop-up parameter references for recognized functions. The IDEs for the Java-based platforms, as well as Microsoft Visual InterDev for ASP, have this feature. PhpEd for PHP has it as well (Yank, 2008).

In accordance with the move towards GUI-based systems, it could be better if the IDE has an HTML toolbox for adding HTML components as well as "What-You-See-Is-What-You-Get" (WYSIWIG) editing. Unlike the IDEs for the other platforms evaluated, only the Microsoft Visual InterDev for ASP features this (MSDN, 2008).

Given that most Web applications involve the use of database, teaching could become easier if the IDE enhances database integration by having database tools such as those for viewing tables, adding connection objects *etc*. Detailed examination of the IDEs shows that this feature is missing in the IDEs for the Java-based platforms, but available in Visual InterDev for ASP (MSDN, 2008), as well as in EasyPHP IDE for PHP. Fuecks (2008) confirms this by warning former VB users not to forget that there is ADOdb which is an ADO equivalent for PHP.

Furthermore, being equipped with context sensitive helps could enable such IDEs to assist students who are "stuck" in certain programming activities. Examination of the various IDEs shows that those for the Java-based platforms have such features, unlike the IDEs for the other platforms evaluated.

Detailed information could be easily accessed if the IDE allows searching documentation and Application Programmers' Interface (API) from the help menu. All the IDEs for the various platforms have such features (JBuilder, 2008; MSDN, 2008; Yank, 2008).

In long statements, it is useful if the IDE shows split views of the long scripts instead of requiring one to scroll and lose sight of beginning portions. It is observed that this feature was missing in the IDEs for the Java based platforms and ASP. PhpEd for PHP does, however, have this feature (Yank, 2008).

While coding and testing developed systems, one often needs to see different views such as the class view, program view, debugging view, and the output view. It is observed that all the IDEs for the various platforms have this feature.

For those functions obtainable via the menu, it would be additionally beneficial if the IDE has configurable keyboard shortcuts. All the IDEs for the various platforms have this feature.

Even after the script is syntactically correct, students often have difficulty in constructing the applicable Uniform Resource Locator (URL) to view the results. A useful feature is IDE's automatic construction of the URL for the output at the click of a button. Unlike the IDE for the Java-based platforms, observation shows that this feature is available in the IDEs for ASP and PHP.

Although it was previously established that it would be useful for novices if the platform is case-insensitive, it would be even better if the IDE has the ability to switch case-sensitivity on and off as necessary. Observation shows that the IDEs for the Java-based platforms have such feature (JCreator, 2008), unlike the IDEs for the other platforms under study.

If a platform is equipped with an independent GUI development tool, a situation where one has to learn and use another language for the front-end design can be avoided. The Java-based platforms have the "Swing" class for the front-end GUI design, unlike the IDEs for the other platforms that use HTML for the front-end GUI design.

Also useful is a *class* wizard which assists in the creation of classes. JBuilder has such facility (JBuilder, 2008) but the other platforms do not.

It is easier to be able to start and stop the server from within the IDE. Only PhpEd for PHP has such facility.

Last but not the least, it would be valuable if an IDE supports an unlimited undo and redo feature so that desired actions can be repeated and unwanted changes can be easily discarded. All the IDEs for the platforms under consideration have 'undo' and 'redo' buttons (JBuilder, 2008; MSDN, 2008; Yank, 2008).

Summary

The application of the established criteria reveals that Java Servlet has the highest score followed by JSP and then ASP and PHP. Although ASP has simpler architecture with good IDE, its drawbacks stem from being largely Microsoft-based, costly and less able to interoperate with other platforms. It should however be noted that the scores are subjectively based on the researcher's knowledge and experience as well as the current design of the platforms. They are thus subject to change with time. Therefore, the emphasis of this study is on the establishment of the criteria that enables the selection among the platforms. Users can thus adapt these criteria to their own taste, associate weights to each criterion and can also test or apply the criteria on other platforms.

6. CONCLUSION

It is important to ensure that the various tasks in web application development are done with adequate ease and in reasonable time. These ideas form the basis of this study to evaluate the ease of use of dynamic web application development platforms. By studying the features of many platforms, features that are desirable for enhancing the ease of use of Web application development platforms are identified. These enabled the establishment of criteria to determine the ease of use of the platforms. The significance of the study lies in the establishment of a comprehensive but specific set of criteria that can be used as a scientific basis for selection. Evaluation of platforms with this framework will contribute towards the determination of suitable platforms for teaching web application development in tertiary institutions.

7. REFERENCES

Adeyemo, A.M. (2000) "Towards a framework for the evaluation of Web sites", *M.Com. dissertation*, University of Pretoria.

Apte, V., Hansen, T. and Reeser, P. (2003) "Performance comparison of dynamic Web platforms", *Computer Communications*. Vol 26, No.8, pp888–898.

Ashenfelter, J.P. (1999) *Choosing a Database for Your Web Site.* John Wiley & Sons, New York.

Bergin, J. (1996) "Object Technology in the Classroom-Java as a Better C++", *ACM SIGPLAN Curricular Patterns,* Vol 1, No.2, pp21-27.

Bowling, A. (2002) *Research methods in health*. 2nd edition, Open University Press, Buckingham.

Cecchet, E., Chanda, A., Elnikety, S., Marguerite, J. and Zwanenpoel, W. (2003). "Performance Comparison of Middleware Architectures for Generating Dynamic Web Content", *Lecture Notes in Computer Science: Middleware,* Vol 2672, No.2003, pp242-261.

Coertze, D. and Heath, R. (1997) *Research Methodology for Technikon Students: A* *practical Approach.* Technikon Natal publishing, South Africa.

Cooper, R. (2001) "Software for managing Web sites" In: *Proceedings of the South African Institute of Computer Scientists and Information Technologists (SAICSIT) Annual conference*. September 2001. Pretoria, South Africa: SAICSIT.

Devdaily. (2009). *Servlet Runner 101: Testing your Java Servlets with the servletrunner* [online], http://www.devdaily.com/java/edu/pj/pj010025/index.shtml

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998) *Human-Computer Interaction*, Prentice Hall, Hertfordshire.

Fuecks, H. *Is PHP the natural upgrade path for ASP 3.0/VB 6?* [online], http://www.sitepoint.com/blog-post-view.php?id=175686

Hartman, H. (2001). Tools for dynamic Web sites: ASP vs PHP vs ASP.NET, *Seybold Report Analysing Publishing Technologies*, Vol 15339211, No.12, pp1-10.

JBuilder. (2008) "*JBuilder 9 Personal Tutorial*", [online], http://www.cs.dal/~arc/teaching/CS2110/TutorialS/introToJBuilderTutorial.htm

Jcreator. (2008) "*Using JCreator: IDE tutorial",* [online],http://www.cs.usm.maine.edu/~welty/cos160JCreatorOnlineTutorial/

Lerdorf, R. and Tatroe, K. (2002) *Programming PHP,* O'Reilly & Associates Inc, Sebastopol, Calif.

Lim, B.L. (2002) "Teaching web development technologies: Past, present, and (near) future", *Journal of Information Systems Education*. Vol 13, No.2, pp117-123.

Marshak, M. and Levy, H. (2003) "Evaluating Web user perceived latency using server side measurements", *Computer Communications, Vol* 26, No.8, pp872-887.

McMillan, J.H. and Schumacher, S. (2001) *Research in education*. 5th edition, Addison Wesley Longman, New York.

MSDN. (2008) "Introducing Visual InterDev", *Microsoft Visual Studio Help file*.

Prechelt, L. (2000) "An empirical comparison of seven programming languages", *Computer*. Vol 33, No.10, pp23-29.

Ross, D., Zymaris, A. and Con, D. (2000) "DB Forms, PHP, MYSQL, and PHPLIB: A reusable database framework for writing Web apps", *Dr. Dobb's Journal: Software tools for the professional programmer*. Vol 25, No.8, pp11-16.

Sebesta, R.W. (1996) *Concepts of Programming Languages,* Addison-Wesley, Reading, Mass.

Singh, S., Erwin, G. and Kotze, P. (2001) "Electronic Business Accepted Practices (e-BAP): Standardised HCI for e-commerce in South Africa", In: *Proceedings of the SAICSIT Annual Postgraduate research symposium*. September 2001. Pretoria, South Africa.

Singh, S. and Kotze, P. (2002) "Towards a framework for e-commerce usability", In: *Proceedings of the annual SAICSIT conference*. September 2002. Port Elizabeth, South Africa.

Sun Microsystems. (1997) "JDBC Guide: Getting Started", [online], http://www.pearson.com.mx/component/component_java_21_days/jdk/1.2/doc/html/guide/jdbc/getstart/introTOC.doc.html

Wiedenbeck, S., Ramalingam, V., Sarasamma, S. and Corritore, C.L. (1999) "A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*", Vol 11, No.3, pp252-282.

Yank, K. (2008) "Review – PHPEd 3.2 (NuSphere)", [online]. http://www.sitepoint.com/print/review-phped-3-2-nusphere

8. APPENDIX

Table 1. Scoring for the platforms based on the criteria on the availability of features of smart IDEs.

	Criteria questions	Servlet	JSP	ASP	PHP
1	Does the platform use interpreters rather than compilers?	1	2	3	3
2	Is the IDE both DOS and Window based?	3	1	1	3
3	Does the IDE automatically create directories to store the project?	3	3	3	1
4	Does the IDE show the directory tree for the project?	3	3	3	3
5	Does the IDE allow the menu-driven addition of Web items such as HTML pages, scripts, style sheets, etc.?	3	3	3	3
6	Does the IDE enhance maintenance by encouraging good readability and writability by way of automatically formatting and indenting codes?	3	3	2	3
7	Does the IDE have hotkey matching of braces, parenthesis, and angle brackets?	3	3	1	2
8	Does the IDE have customizable code formatting templates?	3	3	3	3
9	Is the IDE equipped with tracing and debugging aids to assist students in debugging their codes?	3	3	3	3
10	Does the IDE offer other useful and smart editing facilities such as identifying errors as soon as commands are entered, rather than waiting till compilation time?	1	1	1	1
11	Does the IDE give syntax highlighting with configurable colors?	3	3	3	3
12	Does the IDE have the ability to switch case sensitivity on and off as necessary?	3	3	1	1
13	Does the IDE have function and variable name auto-completion?	3	3	1	2

Scale: 3 = "Yes", 2 = "Not quite or with some workaround", and 1= "No".

Table 2. Scoring for the platforms based on the criteria on the availability of features of smart IDEs continued.

	Criteria questions	Servlet	JSP	ASP	PHP
14	Does the IDE have pop-up parameter references for recognized functions?	3	3	3	3
15	Does the IDE have an HTML toolbox for adding HTML components as well as WYSIWIG editing?	1	1	3	1
16	Does the IDE enhance database integration by having database tools such as viewing tables, adding connection objects?	1	1	2	2
17	Is the IDE equipped with context sensitive helps to assist students who are "stuck" in certain programming activities?	3	3	2	1
18	Does the IDE allow searching documentation and API from the help menu?	3	3	3	1
19	Does the IDE show split views of long scripts?	1	1	1	3
20	Does the IDE show different views such as the explorer class view, program view, debugging view, output view?	3	3	3	2
21	Does the IDE have configurable keyboard shortcuts?	3	3	3	3
22	Does the IDE automatically construct the URL for the output?	1	1	3	3
23	Is the platform equipped with an independent GUI development tool?	3	3	2	1
24	Is there a class wizard to assist in the creating of classes?	3	3	1	1
25	Is it possible to start and stop the server from within the IDE?	1	1	1	3
26	Does the IDE support unlimited undo and redo?	3	3	3	3
	TOTAL	64	63	58	58

Scale: 3 = "Yes", 2 = "Not quite or with some workaround", and 1= "No".

Integrating SAP to Information Systems Curriculum: Design and Delivery

Ming Wang
ming.wang@calstatela.edu
Department of Information systems
California State University
Los Angeles, CA 90032, USA

ABSTRACT

Information Systems (IS) education is being transformed from the segmented applications toward the integrated enterprise-wide system software Enterprise Resource Planning (ERP). ERP is a platform that integrates all business functions with its centralized data repository shared by all the business operations in the enterprise. This tremendous change has led to teaching ERP applications becoming an important issue in information systems education. This paper presents a comprehensive approach to teaching SAP which includes the course design, course delivery and learning outcomes. SAP is utilized as an ERP software tool in the course. The course emphasizes on teaching SAP implementation as a whole process of transforming business procedures to organization wide requirements. The paper concludes with suggestions to those who seek to use SAP as a tool to teach ERP in the area of enterprise system integration.

Keywords: ERP, Enterprise Resource Planning, SAP, IS curriculum, business processes

1. INTRODUCTION

ERP systems are generic and packaged software systems that provide comprehensive functionality and business process integration across the enterprise. These enterprise-wide software systems offer significant potential benefits, as suggested by the growing scholarly literature that seeks to conceptualize and measure types of organizational outcomes, business impacts, and return on investment among ERP adopting enterprises. SAP is an ERP product that has been utilized very popularly in the industry and academics. SAP is complex, learning curve is steep and its implementation is critical and challenging for many SAP educators. Unlike other computer applications, SAP includes the multidisciplinary scope of enterprise system concepts that requires internal cross-disciplinary coordination. Learners must acquire and understand cross functional business processes while implementing the SAP software.

There are a number of efforts that have been made to incorporate SAP implementation into IS curriculum (Davis, & Comeau; 2004; Pellerin, R. and Hadaya, P., 2006). Some SAP educators have responded to the challenges by using case teaching approach to recreate the organizational context within which ERP implementations are conducted. The case study approach usually achieves in more process-oriented thinking than traditional or functional teaching approach does. This approach allows students to develop high-order reasoning skills with hands-on experience (Hackney et al. 2003, Fedorowicz et al. 2004) which in turn increases their motivation and interest in the subject. But, these teaching cases rarely allow students to experience all the challenges of the whole process of changing business processes. Instead of teaching solely on configuration activities, IS educators should focus on teaching topics that reflect the complex reality of SAP implementation.

By recognizing the limitations of case teaching methodology, the authors proposed a comprehensive framework to teaching SAP implementation which includes SAP infrastructure, implementation methodology, process modeling, business process reengineering and configuration and SAP landscape to corresponding SAP hands-on activities. At the end, the authors conducted an evaluation questionnaire to explore learning outcomes and students' reaction to the course. In the conclusion, the authors offer several suggestions to those who seek to teach enterprise integration education. The paper is intended to serve as a useful teaching resource for those IS educators who are interested in teaching ERP/SAP implementation. The remaining paper is organized in four sections: course design, course content, course activities and conclusion.

2. COURSE DESIGN

The SAP Fundamentals course has been taught to senior undergraduates and graduates in the Department of Information Systems for four quarters during the 2007-2008 year. The goal is to teach students to understand business-centric information system and be able to implement SAP software. After successful completion of this course students should be able to:

- Understand SAP system infrastructure, functionality and components
- Reengineer SAP business process procedures to the organization’s requirements
- Utilize process modeling tools to describe and improve business processes
- Configure an SAP system to meet the business requirements in an organization

The proposed framework for teaching SAP implementation is presented in Appendix I. Introduction of each new SAP topic is followed by a set of SAP hands-on activities. ERP business process concepts are based on the contents in the textbooks (Hayen, 2007; Monk & Wagner 2008). SAP hands-on activities are on based on the electronic downloads from the SAP University Alliance Curriculum Innovation Web site.

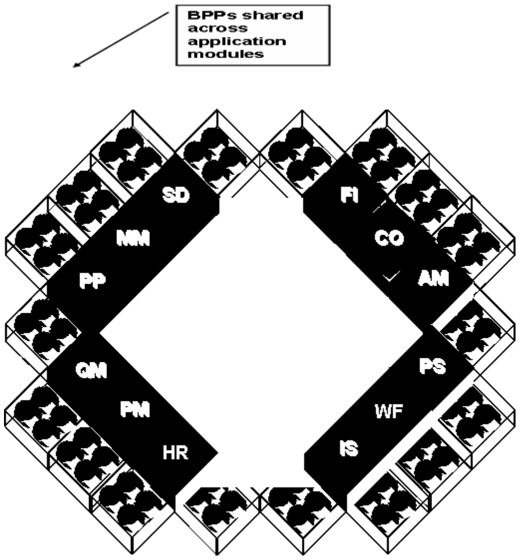
3. COURSE CONTENT

SAP Infrastructure and Components

SAP infrastructure consists of application modules and business process procedures (BPPs). Each application module is a collection of a number of related business process procedures (BPPs), as illustrated in Figure 1. BPPs are the real programs that provide the functionality of the SAP System. Many of these BPPs are used by more than one of the application modules. Business processes and business functionality found in the organization are built with these application modules and BPPs. The twelve application modules are organized into the following four categories.

- The financial management application modules include Financial Accounting (FI), Controlling (CO), Fixed Asset Management (AM) and Project System (PS).
- The logistic modules include Sales & Distribution (SD), Material Management (MM), Production Planning (PP), Quality Management (QM) and Plant Maintenance (PM).
- The Human Resources (HR) module supports human capital management (HCM), payroll and the planning and control of personnel activities.
- Workflow (WF) and Industry Solutions (IS) are known as the Common Systems. The remaining modules are known as primary application modules, and WF integrates the functionality of these modules.

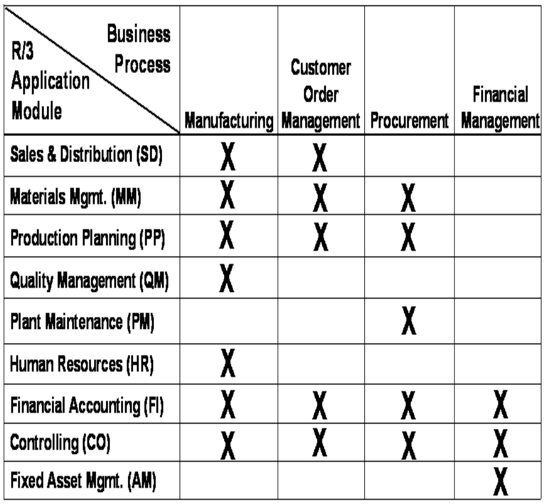
Figure 1 Business process procedures (BPPs) shared across application modules



SAP Enterprise software integrates the core business processes found in an organization. Each business process consists of business application modules. Thus, SAP business processes are accomplished via a very comprehensive set of BPPs available within the various application modules, as illustrated in Figure 2. The following business processes can be used to illustrate the supply chain management processes.

- Manufacturing Planning and Execution.
- Procurement (purchasing).
- Financial/management accounting Customer Sales Order Management

Figure 2 Application modules integrated to business processes



R/3 Application Module \ Business Process	Manufacturing	Customer Order Management	Procurement	Financial Management
Sales & Distribution (SD)	X	X		
Materials Mgmt. (MM)	X	X	X	
Production Planning (PP)	X	X	X	
Quality Management (QM)	X			
Plant Maintenance (PM)			X	
Human Resources (HR)	X			
Financial Accounting (FI)	X	X	X	X
Controlling (CO)	X	X	X	X
Fixed Asset Mgmt. (AM)				X

Business Process Reengineering (BPR)

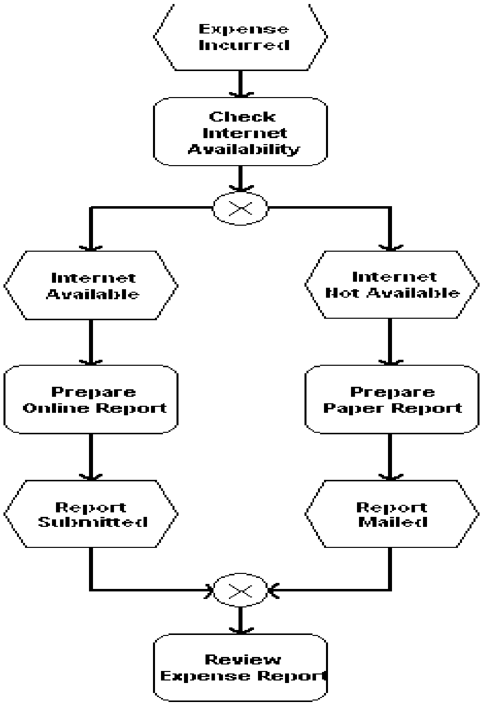
Business Process Reengineering (BPR) transforms organization operations to entirely new and more effective business processes. BPR is primarily cross-functional in its focus. BPR involves questioning assumptions. In the mid 1990s, especially workflow management systems were considered as a significant contributor to improved process efficiency. BPR is usually utilized in the Blueprint Phase.

There are numerous BPR approaches and they each differ according to the magnitude of the change and the change effort involved. Selecting the right model to teach BPR is thus a difficult task. We adopted the model developed by Guha et.al. (1993). This BPR model consists of six phases: 1) Envision new processes, 2) Initiating change, 3) process diagnosis, 4) process redesign, 5) reconstruction and 6) process monitoring.

Business Process Modeling

A process model is a graphical illustration of the business processes. It is a set of communication means that can be utilized by the ERP/SAP team members for BPR in the Blueprint phase. Two types of commonly used graphical representations are the Deployment Flowchart and Event Process Chain (EPC) Diagrams. They can both visually illustrate the process requirements and potential improvements that make the business processes more efficient. The deployment flowchart uses the symbols of basic flowchart representing the flow of goods or data among individual roles or departments of an organization. The deployment flowchart is also referred to as a swimlane chart. Another widely used modeling tool is called the hierarchical modeling flowchart which provides the flexibility of describing processes in more or less details. EPC is another graphical diagram that can be utilized to describe the process driven structure of enterprise systems. Unlike flowcharting, EPC enforces a strict structure using the structure: event-function-event: for an event, its object + past participle; for a function, its verb + object. Figure 3 illustrates an EPC diagram with two XOR connectors showing that a salesperson can submit his/her expense report online **(**Monk & Wagner, 2008).

Figure 3 Event process chain (EPC) diagram

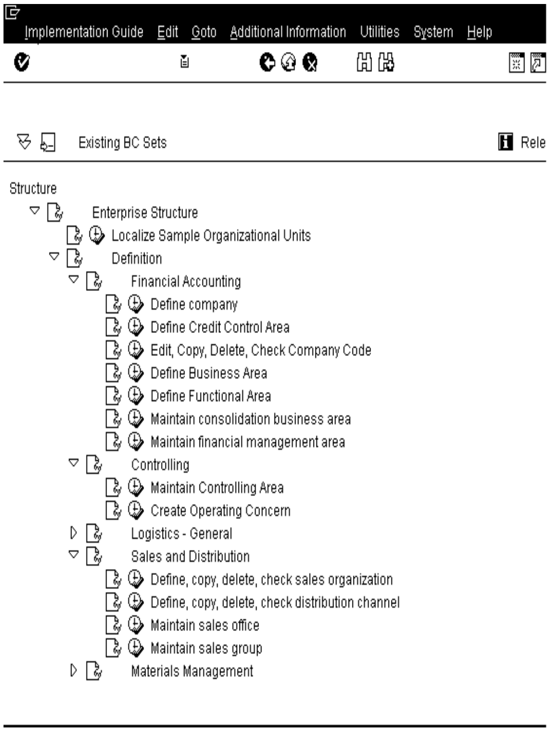


SAP Configuration

The completed reference model is known as the Enterprise Model when the configuration activities have matched the business requirements to the processing available in the Reference Structure. The Implementation Guide (IMG) is a SAP system tool for actually making the settings that configure the R/3 System to meet a company's requirements. The SAP IMG is an integral part of the R/3 System that is used in the configuration of the R/3 System. About 80 percent of a typical SAP system installation is handled by the IMG configuration setting. Another 10 percent are enhancement of ABAP tools and appended structure, while 10 percent are customer developed via ABAP tools.

A Project IMG is a subset of the Reference IMG that contains the documentation for selected IMG components that are implemented as part of the specific configuration project. The Project IMG in Figure 4 facilitates the establishment of the organization arrangement during a company's configuration by using IMG.

Figure 4 SAP Implementation Guide (IMG)



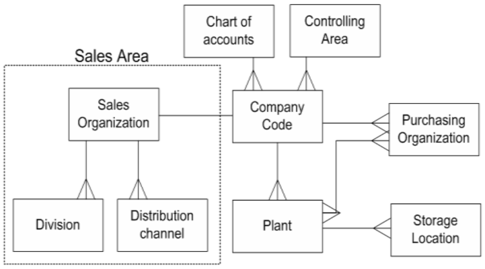
- Structure
  - Enterprise Structure
    - Localize Sample Organizational Units
    - Definition
      - Financial Accounting
        - Define company
        - Define Credit Control Area
        - Edit, Copy, Delete, Check Company Code
        - Define Business Area
        - Define Functional Area
        - Maintain consolidation business area
        - Maintain financial management area
      - Controlling
        - Maintain Controlling Area
        - Create Operating Concern
      - Logistics - General
      - Sales and Distribution
        - Define, copy, delete, check sales organization
        - Define, copy, delete, check distribution channel
        - Maintain sales office
        - Maintain sales group
      - Materials Management

Organization Structure

The SAP Enterprise Structure can be represented by an organizational diagram in Figure 5, which is a model of the relationships among various SAP system organizational elements.

- The organizational diagram represents the organizational structure implemented within the SAP R/3 System.
- This organizational structure is a required prior to any SAP configuration, regardless of the number of applications modules that are implemented for a particular company.
- The available organizational elements are the same whether the SAP R/3 System is being implemented for manufacturing or service related industry.

Figure 5 Organization diagram



ABAP Development Workbench

The ABAP (Advanced Business Application Programming) Development Workbench is an integrated set of fourth-generation tools, which support the implementation of critical client/server applications, or add-ons to standard R/3 System modules. This is where customer developed processes takes place. This toolset is especially suited to R/3 System installations that require enhancements to standard R/3 business applications with customized add-on functionality. The major components of the ABAP Development Workbench include ABAP Programming Language, ABAP Dictionary, ABAP Editor, ABAP Function Library, Data Modeler and R/3 Repository.

ERP Implementation Methodology

Accelerated SAP (ASAP) is a System Development Life Cycle (SDLC) methodology that has been created to guide the rapid implementation of the SAP Enterprise System. It is a computer systems analysis and design methodology and tools provided by SAP AG. This methodology seeks to standardize and expedite the typical SAP implementation. It has proven to be effective when implementing the ERP/SAP solution across industries and different customer environments.

From an academic point of view, the use of the methodology has significant value as it is aligned with industrial standards and procedures defined in Project Management Body of Knowledge (PMBOK) Guide. The *PMBOK Guide* is an internationally recognized standard (IEEE Std. 1490-2003) that provides the fundamentals of project management as they apply to a wide range of projects, including construction, software, engineering, automotive, etc. This guide is consistent with other management standards such as International Organization for Standardization (ISO) 9000 and the Software Engineering Institute's Capability Maturity Model® Integration (CMMI).

In practice, ASAP makes it easier to assign consulting tasks among different projects effectively and to provide an increased level of consistency. The Solution Manager is the SAP platform that delivers the ASAP methodology.

1. Project Preparation – Define the project’s scope. Organize the team, the vendors and hardware.
2. Business Blueprint – Define the business process requirements; discuss configuration, discuss data migration.
3. Realization – Configure the system, write middleware interfaces, develop ABAP code.
4. Final Preparation includes testing and training.
5. Go Live – Begin using the system.

System Landscape

The landscaping is the environment where customization activities of configuration are actually performed, tested and released to production. The configuration activities take place in the Development System. The Development System includes customizing client, testing client unit, sandboxing (prototyping) client and developing customer processes. SAP recommends an organization set up its system for development system, quality testing system, and then production system. So each change to the system, such as software upgrades and releases, can be tested fully before being released into production. The transport directory is the mechanism to load and test these changes. The system landscape in Figure 8 contains all the SAP Systems that are installed for an organization. It consists of three systems, whose SAP Systems are linked by transport routes.

4. CLASS ACTIVITIES

SAP technical hands-on activities are discussed in the following four sections.

SAP Navigation

Navigation in the ERP/SAP system is very important for students who have never used the system before. The most obvious learning need in adoption of an ERP system is to acquire operational capability with the software. Navigation and systems operations are concerned with the actual use of the SAP system that focuses on how to utilize the software and perform business transactions. Navigation activities start with log-on to SAP, creation of multiple sessions, and use of title bar, menu bar, status bar, command field, functions of different icons and help.

Implementation Exercises

The laboratory course materials were adapted from the implementation of an integrated business scenario provided by SAP University Alliance. Fitter Snacker scenario is designed to introduce implementation of integrated business processes through configuration of a small manufacturing with production approach called “make-to-stock”. The case introduces students to learn the basics of business process integration by individually configuring a snack bar company in the SAP environment. The Fitter Snacker Client represents the fictitious Fitter Snacker Company that is used as an example in the textbook ((Monk and Wagner, 2007). Exercises for the Fitter Snacker Client match with examples used in the textbook. Students will configure the FI, MM and CO modules of the SAP system using instructions organized into 9 modules. At the end of these modules, students will produce a printout of their G/L accounts, vendors and purchasing information records, respectively, to demonstrate their progress. After completion of Module 9, students will test their configurations with two sets of transaction exercises called Testing Task 1 (TT1) and Testing Task 1 (TT2). In TT1, students perform a range of accounting transactions as a basic test of their FI and CO configuration. In TT2, students complete the transactions to purchase, receive and pay for raw materials, validating their configuration of the MM module. After TT1 and TT2 have been performed, students will configure the PP module using the instructions in Modules 10 and 11, then test this configuration using TT3 and TT4. In TT3, the students set up and run MRP to create planned orders, while in TT4 the students convert planned orders into production orders, which they process to produce 7 cases of completed NRG-A bars. With both TT3 and TT4, the students produce a printout of the Stock/Requirements list for the appropriate material to validate their configuration. After TT3 and TT4 have been performed, the students will configure the SD module using the instructions in Modules 12 and 13. The SD configuration is validated by TT5, which is a sales order processing exercise that has the students sell snack bars produced in TT4 to a customer configured in Module 13. In TT5, the students hand in a printout of the Document Flow to show they have successfully completed TT5.

Implementation Project
After Fitter Snacker Company is configured, students are assigned to configure the Marshall Muffler Company using the Fitter Snacker documentation modules as a guide and data provided from a preliminary site visit to Marshall Muffler. The primary goal of the configuration project is to produce a customized SAP system that can manage the purchase of raw materials, production and sales of mufflers for the Marshall Muffler Company.

Simulation Game
The Muesli Simulation game provides the SAP environment for students to experience an integrated enterprise system. Students are divided into teams. Each team operates a make-to-stock manufacturing plant that interacts with the suppliers and customers by sending and receiving orders, delivering their products and completing the whole cash-to-cash cycle. Students get the opportunity to make strategic decisions for recipe (product design), sale forecasts, production improvements, timing of production release, At the end of the game, raw materials and finished goods inventory will be sold out at a reduced price. Team members may renegotiate their bank loan with the bank at the end of each quarter. Interest is calculated quarterly and will be charged automatically at the end of a quarter. Customers do not change through out the game, hence you may learn from past their behaviour. The company (team) displaying the highest equity value
(shareholder capital) at the end of the simulation wins the game. As a make-to-stock manufacturing company, there are four of processes that must be performed (i) the planning process, (ii) the procurement process, (iii) the production process, and (iv) the sales process. Each process can be decomposed into transactions. To complete the four operational processes mentioned above, a total of fourteen transactions must be performed. Most of the transactions involved in these processes are operational in nature, such as to purchase materials or to deliver finished products (ERPsim, 2008). The four operational processes with fourteen transactions are illustrated in Appendix II.

5. CONCLUSION

The goal of the paper is to propose a comprehensive teaching approach which ties SAP fundamental knowledge to SAP implementation activities. Hands-on experience learning enables students to navigate in some areas of the SAP system, execute tasks, and learn some SAP technical skills, but provide no understanding of why tasks are being performed. The laboratory manuals focus on step by step instruction on task completion, not on business logic (Scott and Sugar, 2004), Thus, hands-on learning experience has limited value unless it reinforced with the in class discussion and review. The combination of hands-on laboratory learning and business process concepts learning via reading, discussion, and case study is powerful and effective approach to learn SAP fundamentals. It is a challenge to balance the two streams and relate the laboratory learning with business process integration in the course.

The purpose of the paper is to shift the teaching focus of SAP implementation from SAP technical issues to organization transformation process. The proposed framework takes into consideration of the various disciplines and different types of competencies needed to effectively transform business processes in an organization. Learning by doing activities make students motivated to learn new materials by themselves. One of the most important features of the comprehensive framework is that it focuses students' attention on the organization transformation process rather than on the technical aspects of SAP systems only. The paper highlights the needs to tighten the SAP laboratory manuals to the business logic so that the students are able to understand why they are learning the SAP laboratory materials and how the system actually works, rather than simply following the laboratory manuals and learning to operate on the computers.

6. ACKNOWLEDGEMENTS

The authors would like to thank Professor Roger Hayen, Professor Ellen Monk and Professor Bret Wagner for their ERP/SAP textbooks. We are

also grateful to the SAP University Alliance Program for providing us SAP software, server hosting and high quality SAP laboratory materials.

7. REFERENCES

Davis, Charles and Jana Comeau, (2004) “Enterprise Integration in Business Education: Design and Outcomes of a Capstone ERP-based Undergraduate e-Business Management Course”. Journal of Information Systems Education, 15 (3), pp. 287-99.

ERPsim 2008 Tutorial Web Site: http://web.hec.ca/erpsim/tutorialerpsim

Fedorowicz, Jane; Ulric J. Gelinas; Catherine Usoff,; George Hachey (2004) “Twelve Tips for Successfully Integrating Enterprise Systems Across the Curriculum”. Journal of Information Systems Education, 15 (3), pp. 235-44

Guha, S.; Kettinger, W.J. and Teng, T.C., (1993) “Business Process Reengineering: Building a Comprehensive Methodology”, Information Systems Management

Hackney, R., McMaster, T. and Harris, A. (2003) “Using cases as a teaching tool in IS Education” Journal of Information Systems Education v. 14 (3), pp. 229-234

Hayen, Roger. SAP R/3 Enterprise Software: an Introduction, McGraw-Hill, 2007

Monk, Ellen and Bret Wagner, Concepts in Enterprise Resource Planning, 3rd.Course Technology, 2008.

Nelson, Robert J., and Ido Millet (2001), "A Foundation Course in ERP and Business Processes: Rationale, Design, and Educational Outcomes." Proceedings of the Seventh Americas Conference on Information Systems, Miami, pp. 992-997.

Noguera, José H., & Edward Watson (2004), "Effectiveness of Using an Enterprise System to Teach Process-Centered Concepts in Business Education." Journal of Enterprise Information Management, 17 (1), pp. 56-74.

Pellerin, Robert and Pierre Hadaya, (2006) “An innovative framework for terracing reengineering and ERP implementation concepts”, Innovation Monograph II: Enterprise resource planning: Teaching and research, pp 13-20.

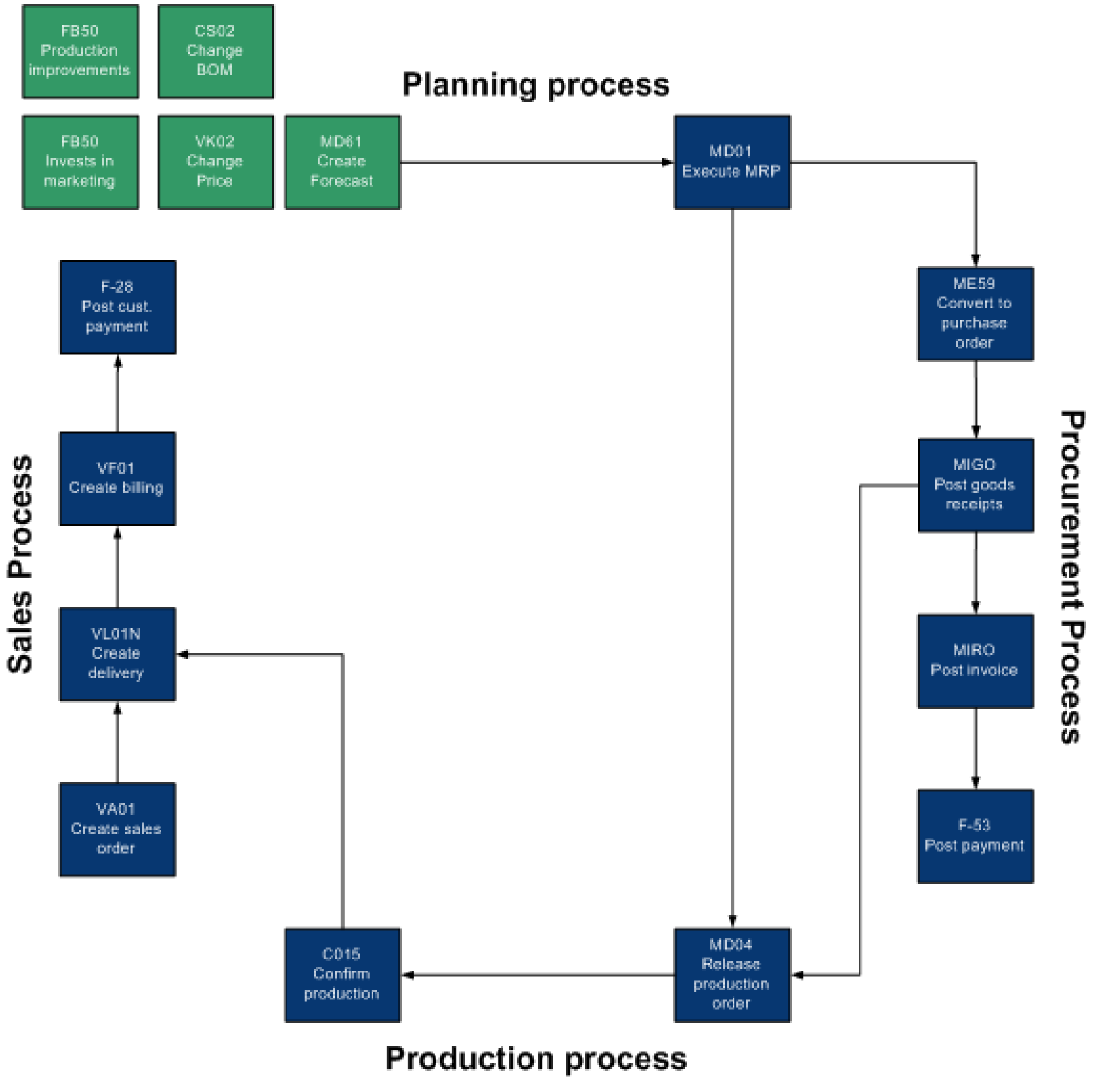
Seethamraju, Ravi C. (2002), "Trials and tribulations in the design and delivery of innovative business course in enterprise resource planning in Australia," Proceedings of the 17th Annual Conference of the International Academy for Information Management (IAIM), Barcelona, pp. 237-249.

Scott, Judy E. and Don Sugar (2004), “Perceived Effectiveness of ERP Training Manuals”. Proceedings of the Tenth American Conference on Information systems, New York, pp. 211-3215

Appendix I Proposed framework for teaching ERP/SAP implementation

ERP Business Process Concepts	**SAP Hands-on Technical Activities**
ERP infrastructure and components	SAP interface navigation
ERP Core business processes	Application modules & business process procedures implementation
Business process reengineering (BPR)	Application modules Process modeling diagrams
Implementation methodology roadmap	Accelerated SAP
Enterprise model and organization structure	Fitter Snacker Configuration: Implementation Guide (IMG)
General ledger and controlling	Fitter Snacker Configuration: financial accounting
Organization structure	Business area customization
ERP Landscape systems	Testing Fitter Snacker configuration
Integrated business operations	ERPSim Simulation game

Appendix II Four Integrated business processes with 14 transactions (ERPSim, 2008)



A Validation Study of Student Differentiation Between Computing Disciplines

Michael Battig
mbattig@smcvt.edu
Computer Science & Information Systems
Saint Michael's College
Colchester, VT 05439, USA

Muhammad Shariq
mshariq@auaf.edu.af
Information Technology & Computer Science
American University of Afghanistan
Kabul, Afghanistan

Abstract

Using a previously published study of how students differentiate between computing disciplines, this study attempts to validate the original research and add additional hypotheses regarding the type of institution that the student resides. Using the identical survey instrument from the original study, students in smaller colleges and in different cultural contexts are studied. Both the original and the validation study consider computing and non-computing majors. Although the original research was largely validated through some strikingly similar results, some significant differences were observed depending on the size and orientation of the institution. Furthermore, we noted some differences in college students outside of the US.

Keywords: information systems, information technology, computer science, software engineering, computer engineering, student perceptions, information systems education

1. INTRODUCTION

The January 2010 issue of ACM SIGSOFT Software Engineering Notes devoted the monthly Software Engineering Education column to the topic of how well students differentiate between computing disciplines (Ardis & Henderson, 2010). The column urged readers to obtain and read a study presented at SIGCSE 2009 (Courte & Bishop-Clark, 2009) that surveyed undergraduate students (both computing and non-computing majors) to determine their understanding of the five major computing disciplines: Computer Science (CS), Information Technology (IT), Information Systems (IS), Computer Engineering (CE), and Software Engineering (SE).

Both articles acknowledged that "there is very little difference between majors and non-majors in these responses." Furthermore, Ardis & Henderson noted that "of the five disciplines Software Engineering appears to be the least understood." These observations piqued our curiosity and led us to engage in several tasks.

We wanted to validate this study through our own administration of the same survey. We contacted the authors of the 2009 SIGCSE study and requested a copy of and permission to utilize their survey instrument. The request was immediately granted and the survey instrument provided to us electronically.

In addition to validating the original study, we were particularly interested in hypotheses related to the understanding of IS and IT in various contexts. Therefore, our sample populations vary slightly from the original studies in that our student subjects reside in different academic and cultural contexts.

The import of this research lies in our continuing efforts to clarify the nature of our discipline so as to attract bright individuals. We do this because "perception problems carry over to computing professionals and educators who, to some degree, have difficulty defining, describing and explaining these disciplines. This, in turn, carries over to prospective students and their parents" (Ardis & Henderson, 2010). Furthermore, "computer applications are found in almost every academic discipline, and the creation of useful, innovative computer applications in any discipline requires both knowledge of that discipline and knowledge of computing" (Walker and Kelemen, 2010).

Therefore, we as educators must continue to explore the verbiage and effective communication necessary to convey to the world outside of our discipline exactly what it is we do and why. This survey is a significant part of examining and improving that endeavor of communicating the essence of our discipline in the world of ideas.

2. SURVEY & HYPOTHESES

An overview of the original survey from Courte & Bishop-Clark, which we used without editing in any way to preserve the integrity of our study, is presented in Appendix A. We encourage readers to read this overview in order to become familiar with the verbiage used to differentiate between the disciplines in the study.

In addition to validating the general results concerning differentiation between majors and non-majors, we were keenly interested in discovering if results at our institutions would match the original findings. Our institutions fit into the small to medium college classification, whereas the original study was conducted at larger universities. Thus, we wanted to test the hypothesis that students at smaller institutions would more accurately differentiate between computing disciplines. We also wanted to add an international dimension to our study. Therefore, we administered the survey to students at the American University of Afghanistan. Our hypothesis was that students at AUAF would less accurately differentiate between computing disciplines than their American counterparts. We also hypothesized that computing majors at both institutions would more accurately differentiate between computing disciplines than the non-computing majors.

3. RESULTS

During the spring semester of 2010, students were given the survey in Business (non-major) and IT, IS, and CS courses at our institutions. Many of the original study surveys were completed electronically during class time. All of our surveys were on paper.

The original study had 375 students responding. The majority (67%) were male. They also had a slight majority who were majors (53%). Our study had 196 usable respondents completing the survey. Some were discarded as unusable (e.g., if the student indicated "don't know" for every answer). The majority of our American respondents were male (68%) and the majority of the Afghan respondents (77%) were male. We had far fewer majors in our survey (28%) than the original.

The survey asks each respondent three questions about each sub-discipline (the order is mixed so as to conceal the *correct* response). Thus each survey contains 15 questions in order to cover all five sub-disciplines. The questions are reproduced in each subsection below. Respondents may indicate one of six answers: CE, CS, IS, IT, SE, or Don't Know. We will contrast the top answers to each question from the original study with our study.

Computer Engineering

Table 1 shows the results for the CE questions. The original study and this validation study produced strikingly similar results for the sub-discipline of Computer Engineering. Each cell contains three lines of data – the first indicating *total students*, the second indicating *majors* and the third indicating *non-majors*. For example, the second question, Builds hardware devices such as iPods, 82% of majors picked CE in the original study, compared to 92% of majors in the US and 56% in Afghanistan. The **boldface numbers** indicate that the percentage is the top response and the correct one. The most frequently answered sub-discipline is listed in the lower portion of cells where the percentage is not bolded. For example, majors in the US picked CE 15% of the time for Question 3 in Table 1 where their top (wrong) answer was CS. The actual percentage that picked CS (and other wrong answers) is shown in Appendix B.

Questions 1 and 2 for CE proved to be validated in our study. Students are generally aware of the fact that Computer Engineers design and build hardware. The results are somewhat exaggerated in that our US population had much stronger results for the majors and the Afghan population had weaker results. The third question is more interesting in that all three populations missed the mark. The top answers for all three populations turned out to be Computer Science, showing that students are universally confused about the fact that Computer Engineers more than Computer Scientists do a fair amount of hardware/software integration.

Table 1 – Computer Engineering Questions and Results (in each cell, the first line indicates total students, the second line indicates the majors, and the third line indicates non-majors)

	Original	US	Afghan
Designs hardware to implement communications systems	**62%** **72%** **51%**	**53%** **82%** **39%**	**38%** 25% **41%**
Builds hardware devices such as iPods	**71%** **82%** **59%**	**72%** **92%** **61%**	**46%** **56%** **44%**
Integrates computer hardware and software	25% 28% 21% **CS**	9% 15% 6% **CS**	14% 13% 14% **CS**

Computer Science

Table 2 shows the results for the CS questions. Once again, the prior study was validated with striking consistency. There are some small variations within the Afghan survey data which will be discussed later. What is striking about these results is how consistently students fail to differentiate between CS and SE. As Ardis and Henderson (2010) have stated, this can "be viewed as a perception problem for all students regarding software engineering."

Table 2 – Computer Science Questions (total students, majors, and non-majors)

	Original	US	Afghan
Uses new theories to create cutting edge software	33% 40% 26% **SE**	18% 21% 17% **SE**	14% 13% 14% **SE**
Focuses on the theoretical aspects of technology	**51%** **56%** **46%**	**57%** **62%** **55%**	25% **31%** 23% **IS**
Utilizes theory to research and design software solutions	36% 40% 31% **SE**	25% 41% 17% **SE**	14% 13% 14% **SE**

We do not believe that these results can be totally attributed to perception confusion between CS and SE. The reader will notice that both questions 1 and 3 in Table 2 feature the word "software" prominently. One wonders how the results would have differed by substituting the phrase "computing system" for "software" in these questions, which would shift the emphasis to *theory*.

Software Engineering

Table 3 shows the results for the SE questions. We have placed the SE and CS sections together since they really tell the same story as the results show. Students are not able to identify the statements in this survey as pertaining to SE. The only exception to this being that US students in our study were able to correctly choose SE related to "testing large scale systems." Again, one wonders how the wording of the survey questions might impact the outcomes. Clearly the word "software" substituted for "technological" or "systems" could change the results even though the emphasis of this line of questioning is on "large scale." In most cases it appears that students make their selection based on certain key words. We can't help but notice that the word "system" occurs in questions 1 and 2 in Table 3, which apparently leads students to choose IS as their answer.

Table 3 – Software Engineering Questions (total students, majors, and non-majors)

	Original	US	Afghan
Focuses on large-scale systems development	12% 17% 6% **IS**	13% 18% 10% **IS**	4% 6% 3% **IS**
Designs testing procedures for large-scale systems	23% 27% 18% **IS**	**22%** **36%** 14%	18% 19% 17% **IS**
Manages large scale technological projects	10% 13% 6% **IS**	7% 18% 1% **CE**	13% 13% 13% **IT**

Information Systems

Table 4 shows the results for the IS questions. Notice that the US students in our survey outperformed the original survey students by about 10 points in each category. The Afghan students on the other hand were more torn between IS and IT when it came to identifying the "business" orientation of IS. We will discuss potential reasons for this below.

Table 4 – Information Systems Questions (total students, majors, and non-majors)

	Original	US	Afghan
Is business oriented	**51%** **59%** **42%**	**59%** **69%** **53%**	29% **31%** 28% **IT**
Combines knowledge of business and technology	**40%** **48%** 31%	**51%** **59%** **47%**	26% 25% 27% **IT**
Selects computer systems to improve business processes	**43%** **49%** **37%**	**47%** **54%** **44%**	**28%** **38%** **25%**

Information Technology

Table 5 shows the results for the IT questions. The results show that the US students in our survey performed comparably to the original in questions 1 and 3, but were more prone to see question 2 as pertaining to CS. The Afghan student responses were mixed between IT, IS and CE (the CE responses pertaining particularly to question 3).

Table 5 – Information Technology Questions (total students, majors, and non-majors)

	Original	US	Afghan
Troubleshoots and designs practical technical applications	**32%** **36%** **27%**	**35%** **38%** **32%**	**23%** **44%** 17%
Applies technology to solve practical problems	**35%** **40%** **30%**	24% 15% **29%** **CS**	**31%** **38%** **30%**
Applies technical knowledge for product support	**55%** **67%** **41%**	**53%** **72%** **43%**	15% **31%** 11%

4. CONCLUSIONS

Our first hypothesis was that students at smaller liberal arts based institutions would more accurately differentiate between computing disciplines. The rationale for this hypothesis is that our smaller institutions emphasize the liberal arts and as such tend to discuss the points of commonality and difference between disciplines (not just limited to *computing* disciplines). As noted by Walker and Kelemen (2010), "Overall, a liberal arts program emphasizes general knowledge, multiple perspectives, alternative ways of thinking, and connections among disciplines." This hypothesis proved to be true among the US computing majors who out performed the students from the original study in 12 out of 15 questions. The non-majors, however, only outperformed the subjects in the original study in 8 out of 15 questions.

Our second hypothesis was that students at AUAF would underperform the American students. Our rationale being that the computing educational infrastructure in Afghanistan is very new. This hypothesis appears to be substantiated by the data in that both majors and non-majors in our study only outperformed the original study in 1 out of 15 questions. We should mention here that our study in Afghanistan only included 16 majors of the 80 total surveys from AUAF. Additionally, out of these 16 majors, 8 were freshman, 6 were sophomores and only 2 were juniors. AUAF is a still in its infancy with no student in the senior year of a computing major.

We would be remiss if we did not point out a particular bias of the Afghan students in this study. The majors at AUAF are in a program that is a hybrid of IS, IT and CS. Many of the students work full or part-time as IT/IS employees with local firms in Kabul. However, they have communicated to the administration on several occasions that they wish to have the department called "computer science" since they perceive that the most prestigious institutions in the US have computer science programs. This desire to be called computer science may significantly influence the students' ability to differentiate between the computing disciplines.

Our third hypothesis was that majors would more accurately differentiate between computing disciplines than non-majors. Kurkovsky (2007) discussed and suggested a need to clear misconceptions among the non-majors about CS. Our study confirms not only the initial finding of misconceptions about CS among non-majors but about all computing disciplines in general. In the original study, the majors outperformed the non-majors in all 15 questions. The US majors in our study outperformed the non-majors in 14 out of 15 questions (the one exception being the 2nd IT question). The Afghan majors in our study outperformed the non-majors in 9 out of 15 questions. Thus, in the US population, we can confidently assert that our original hypothesis was correct, especially given that in many instances the difference was in excess of 20 percentage points. However, the outcomes in the Afghan population were more mixed. What impact (at AUAF) does having the majority of computing major students (88%) in their initial years of studies? This may be an area of further research.

5. REFERENCES

Ardis, M., & Henderson P. (2010). Software Engineering Education. ***ACM SIGSOFT Software Engineering Notes***, 35(1), 4-5.

Courte, J., & Bishop-Clark C. (2009). Do Students Differentiate Between Computing Disciplines? ***Proceedings of the 40th ACM technical symposium on Computer science education***, 41(1), 29-33.

Kurkovsky, S. (2007), Making Computing Attractive For Non-Majors: A Course Design, 22(3), 90-97.

Walker, H. M., & Kelemen, C. (2010). Computer Science and the Liberal Arts: A Philosophical Examination, **ACM Transactions on Computing Education (TOCE),** 10(1).

Appendix A

Overview of the original survey by Courte & Bishop-Clark (2009):

Respondents were presented with phrases descriptive of each discipline and asked to select one discipline that they thought best matched the phrase (options included Computer Engineering, Computer Science, Information Science, Information Technology, Software Engineering, and Don't know). There were 15 phrases with 3 for each discipline, randomly mixed. The phrases were created by surveying colleagues and other practitioners for keywords thought to represent the discipline. Keywords that had a greater consensus were then used to construct the survey phrases.

For **Computer Engineering**, the keyword was *hardware*:

- Designs *hardware* to implement communications systems
- Builds *hardware* devices such as iPods
- Integrates computer *hardware* and software

For **Computer Science**, the keyword was *theory*

- Uses new *theories* to create cutting edge software
- Focuses on the *theoretical* aspects of technology
- Utilizes theory to research and design software solutions

For **Information Science**, the keyword was *business*:

- Is *business* oriented
- Combines knowledge of *business* and technology
- Selects computer systems to improve *business* processes

For **Information Technology**, the keywords were *practical* and *applied*:

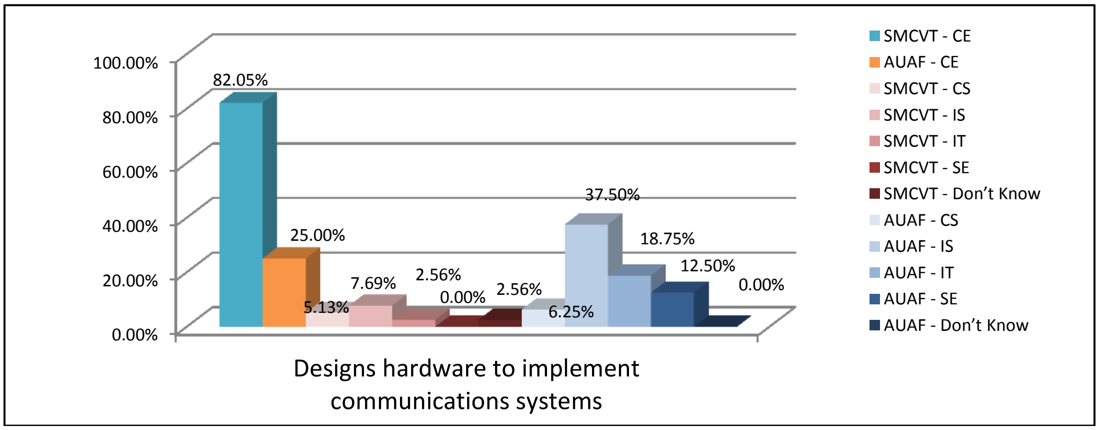
- Troubleshoots and designs *practical* technical *applications*
- *Applies* technology to solve *practical* problems
- *Applies* technical knowledge for product support

For **Software Engineering**, the keywords were *large-scale systems* and *projects*:

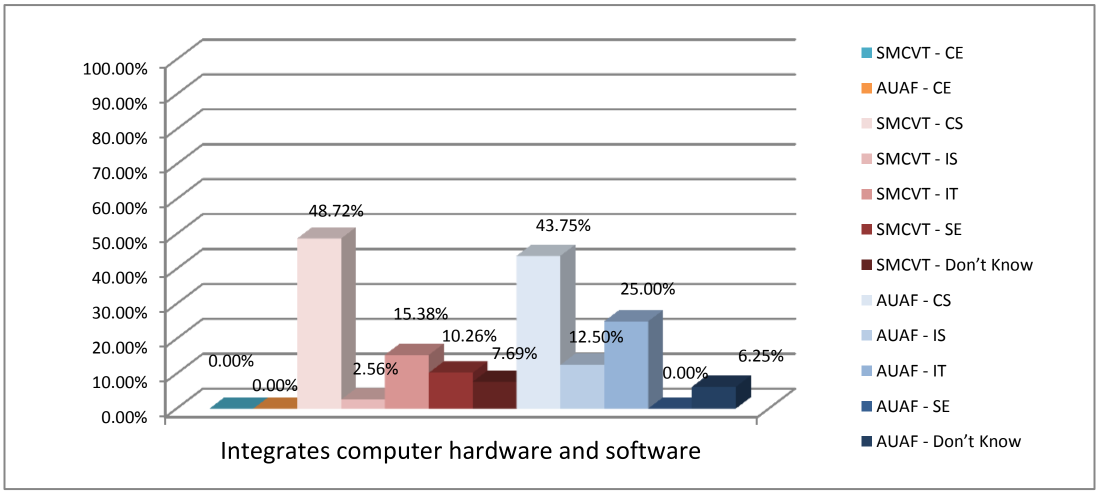
- Focuses on *large-scale systems* development
- Designs testing procedures for *large-scale systems*
- Manages *large scale* technological *projects*

Respondents were also asked to provide basic demographic information, such as gender, age, state and type of employment, student status, major, and type of university.

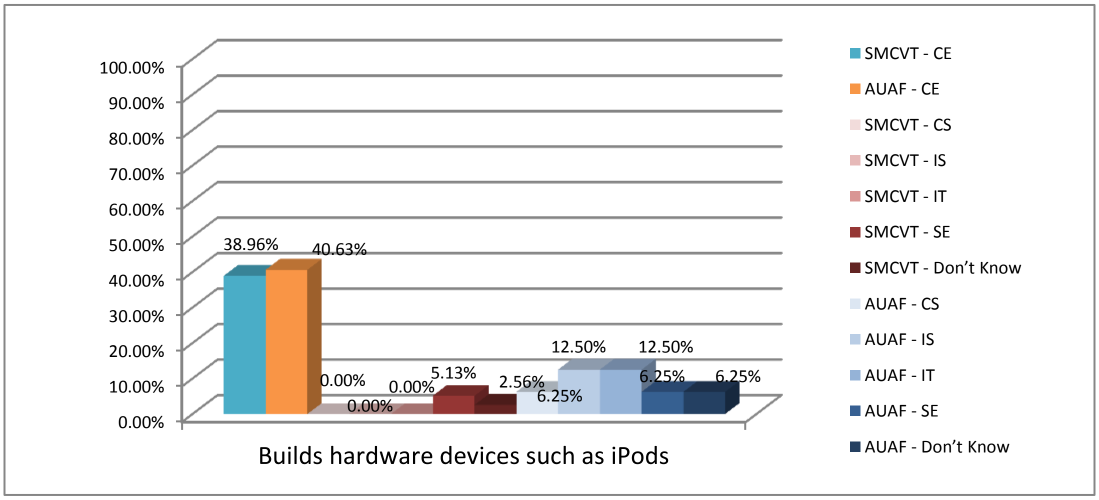
Appendix B - Computer Engineering



Designs hardware to implement communications systems

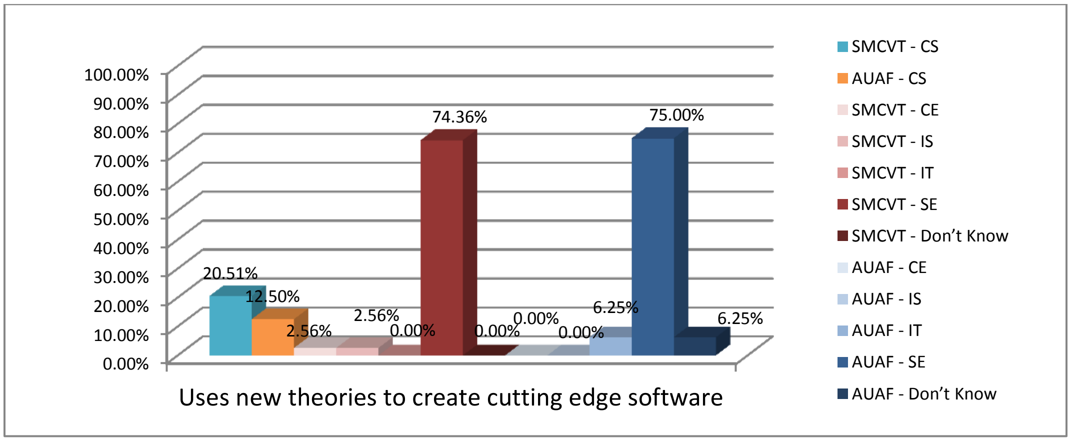


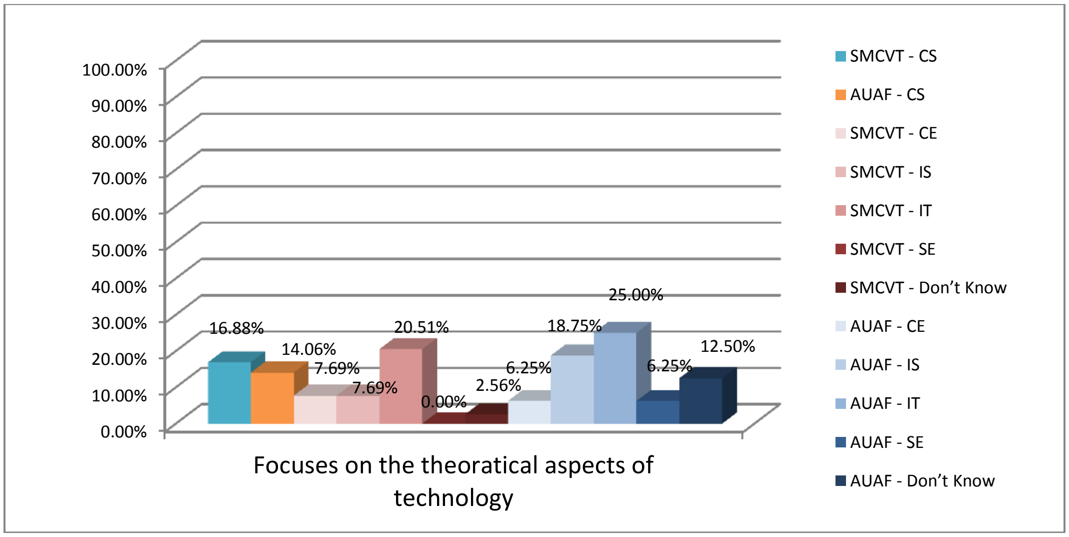
Integrates computer hardware and software

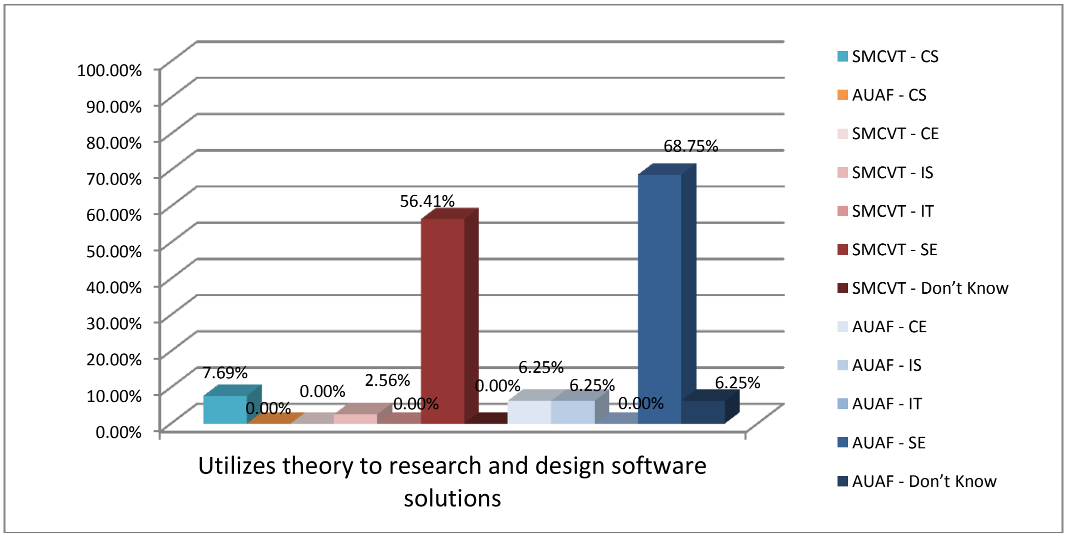


Builds hardware devices such as iPods

Computer Science

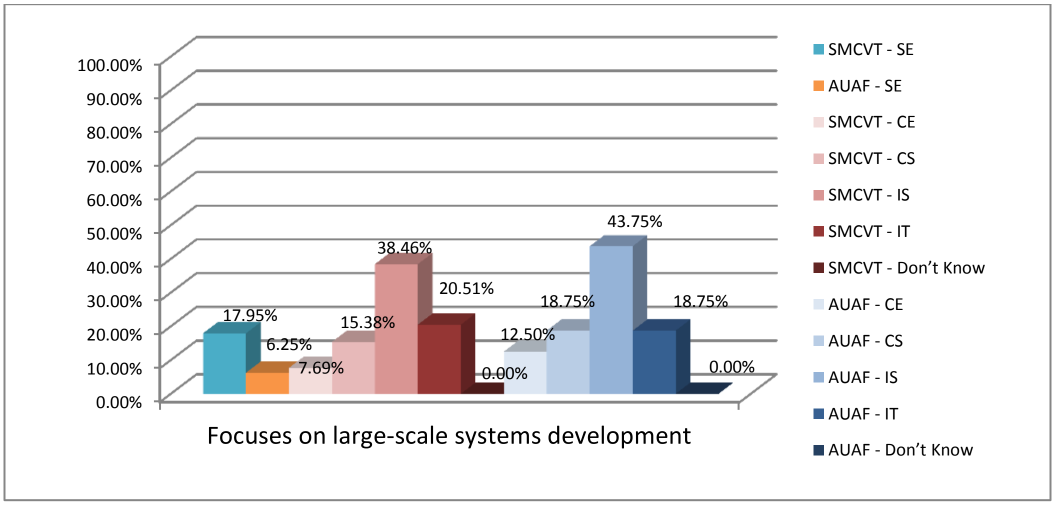


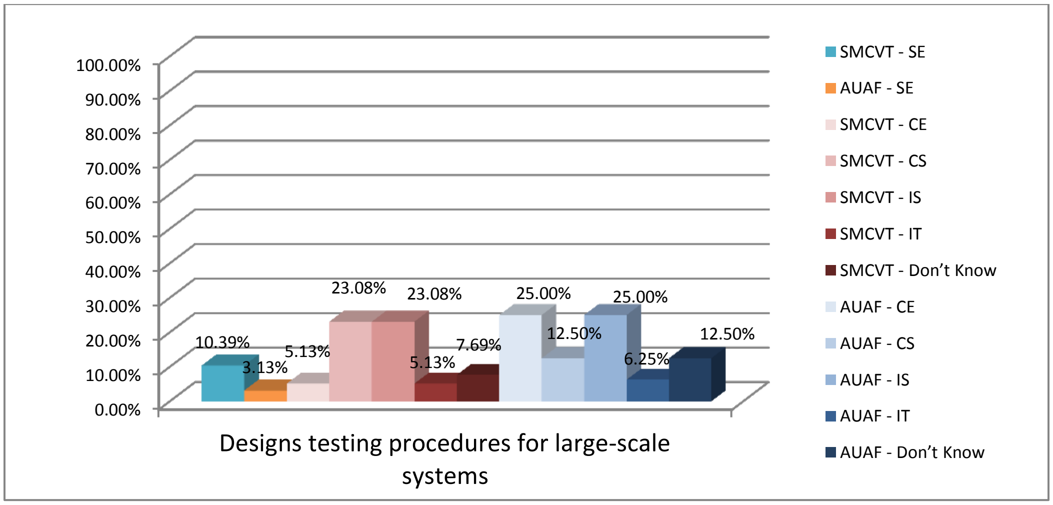


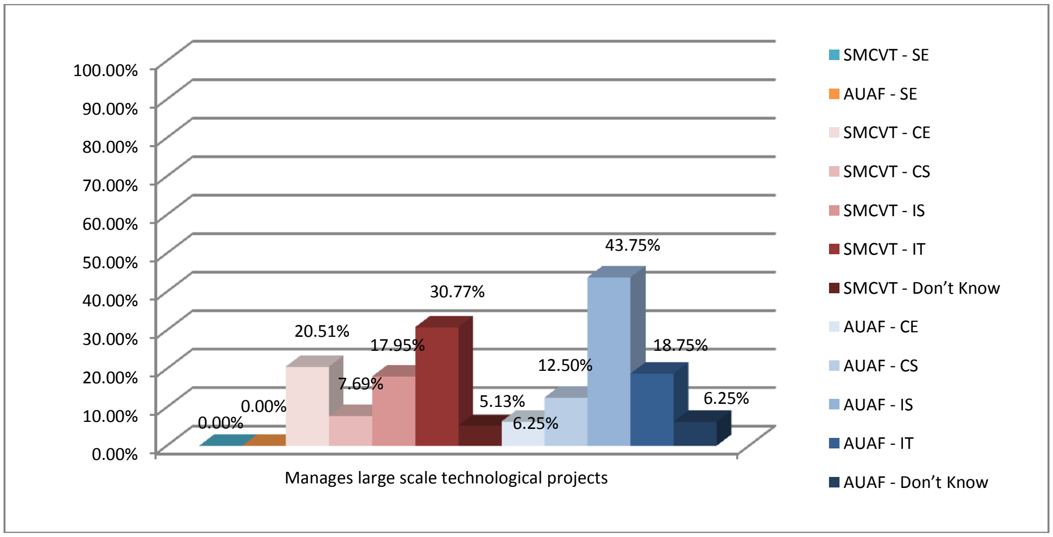


©2011 EDSIG (Education Special Interest Group of the AITP)

Software Engineering

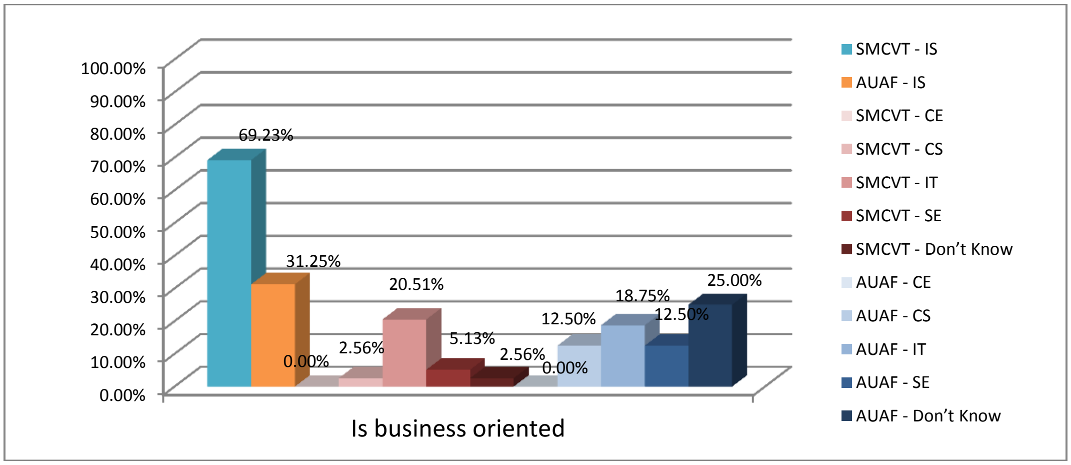


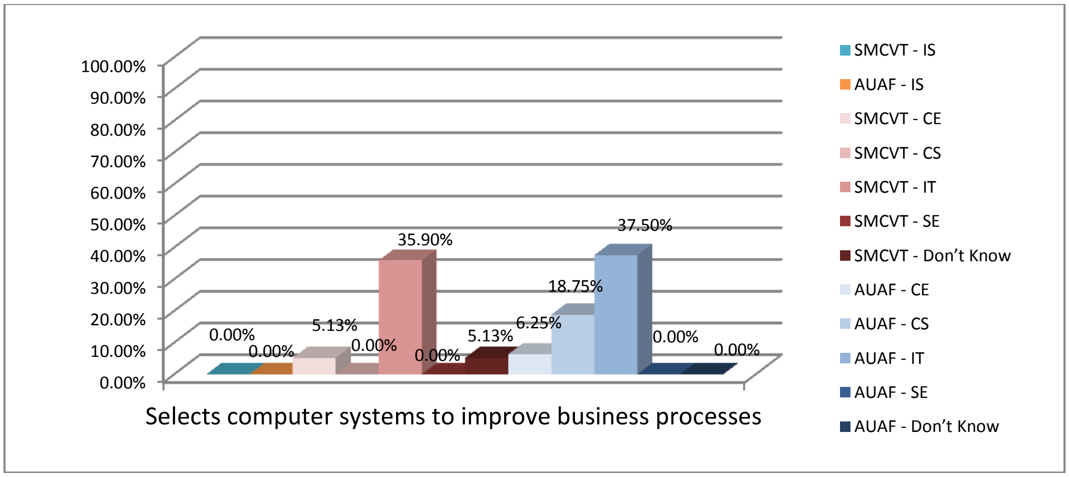


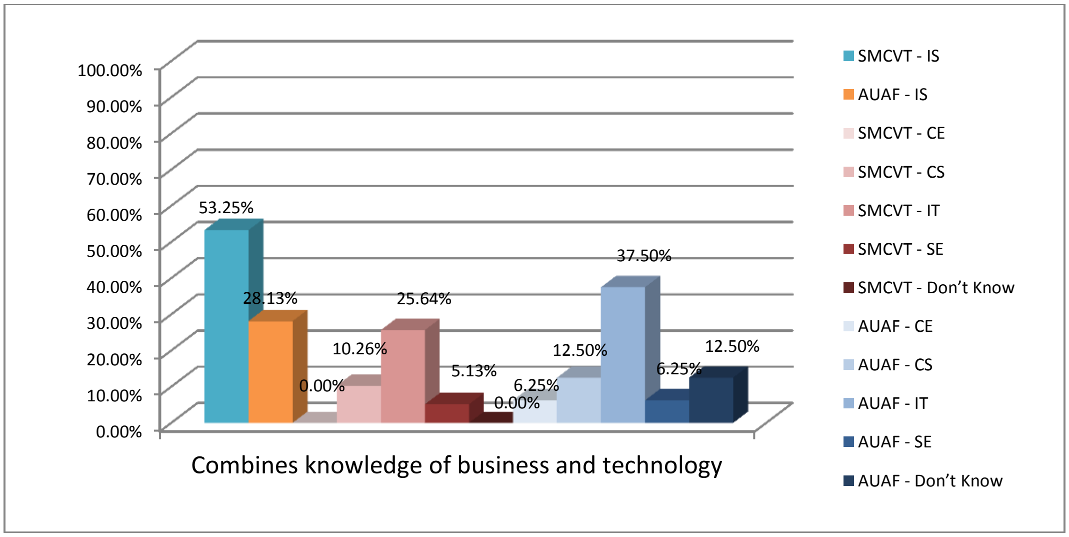


©2011 EDSIG (Education Special Interest Group of the AITP)

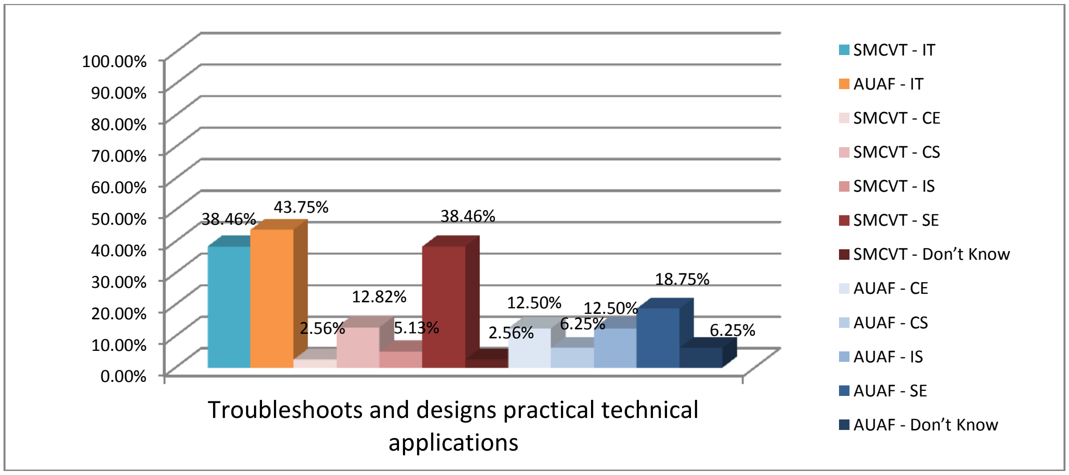
Information Systems



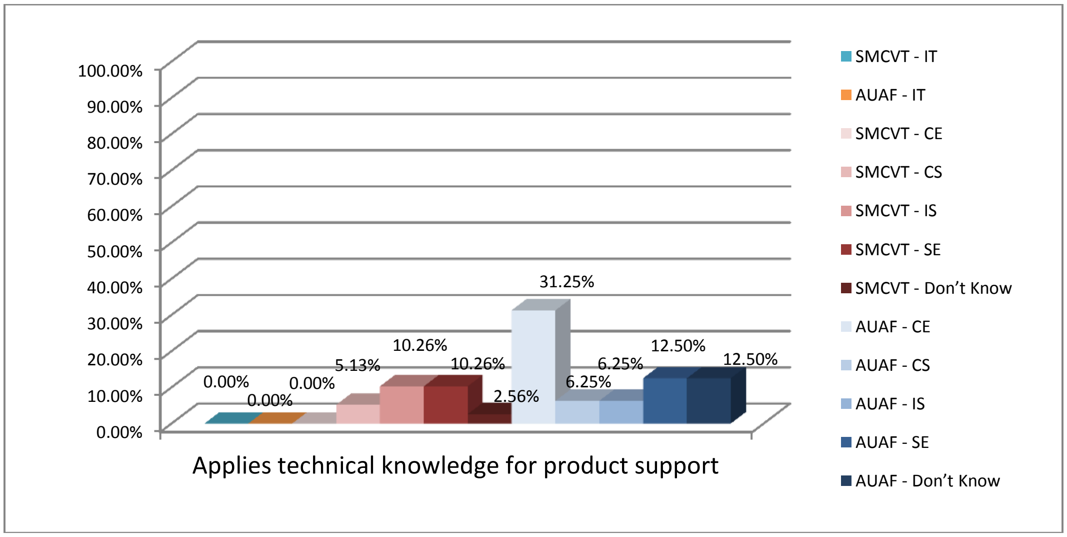




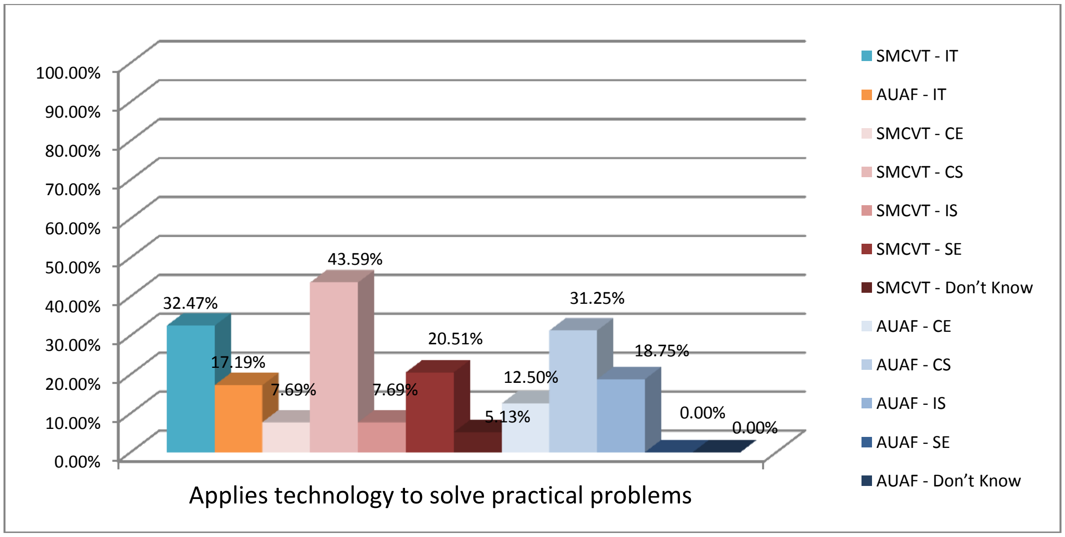
Information Technology



Troubleshoots and designs practical technical applications



Applies technical knowledge for product support



Applies technology to solve practical problems