Volume 14, No. 4
July 2016
ISSN: 1545-679X

Information Systems Education Journal

In this issue:

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published reviewed published by **ISCAP,** Information Systems and Computing Academic Professionals. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCon (http://www.edsigcon.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 under 40%.

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. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

2016 AITP Education Special Interest Group (EDSIG) Board of Directors

INFORMATION SYSTEMS EDUCATION JOURNAL

The Design and Evaluation of Class Exercises as Active Learning Tools in Software Verification and Validation

Peter Y. Wu
wu@rmu.edu
Department of Computer and Information Systems

Priyadarshan A. Manohar
manohar@rmu.edu
Department of Engineering

Sushil Acharya
acharya@rmu.edu
Department of Engineering

Robert Morris University
Pittsburgh, PA 15108, USA

Abstract

It is well known that interesting questions can stimulate thinking and invite participation. Class exercises are designed to make use of questions to engage students in active learning. In a project toward building a community skilled in software verification and validation (SV&V), we critically review and further develop course materials in the software engineering curriculum for an undergraduate course in SV&V. The project involves the joint effort with many other academic institutions and industry partners. There are four topic areas of Software Engineering in our focus: Requirements Engineering, Software Review, Configuration Management, and Testing. We see class exercises as active learning tools for the students in our flipped classroom approach. We present our design of the class exercise: in its generic components envisioning how it may be used in general, but also its use in selected examples to illustrate these components. The class exercise design includes the learning objectives to indicate how the course design meets the learning outcome objectives for ABET accreditation. We further applied a classification of the learning objectives to present the case of the class exercises as active learning tools. Our initial implementation is ready and we are in the process of implementation with partner institutions for feedback and review.

Keywords: Software Verification and Validation, SV&V, Active Learning Tool, Software Engineering.

1. INTRODUCTION

Teaching software engineering at the college level requires balance between the knowledge and hands-on experience. This is especially the case at the undergraduate level when students have just acquired proficiency in programming but are generally short on the appreciation of the practice of software development processes in industry. Funded by an NSF-TUES Grant (National Science Foundation: Transforming Undergraduate Education in STEM), our project focuses on developing new tools for teaching software verification and validation (SV&V) at the undergraduate level. The basic objective of the project is to enhance the quality of software engineering education by increased student engagement in learning as well as bridging the gap between the theoretical knowledge discussed in the classroom and the complexity of real world problems. This endeavor will promote SV&V awareness and increase SV&V practitioners skilled in the practice. The goal is to improve product and process quality levels throughout the software development community, resulting in larger and better skilled SV&V community. Section 2 will briefly describe the background, scope and rationale of the research project.

The project is carried out through an academic-industry partnership. The entire team involves 2 academic development partners, 5 industry development partners and 12 academic implementing partners. In the project, we join with the academic and industry partners to critically review the existing course materials against current methods and best practices. We then refine and further develop the course to address the gaps and inadequacies. To engage the students in active learning, we practice a flipped or inverted classroom approach (Strayer 2012; Bishop & Verleger 2013; Frydenberg 2013) and basically use the class time for activities requiring the students to review lecture and other reading materials beforehand. During the class time, we apply active learning tools to engage the students. Section 3 will describe the specific topic areas of focus in SV&V and the tools we developed.

In this paper we present the class exercises, primarily for class discussion in concert with the other learning tools. Section 4 will describe the design of the class exercise, generically for use in different settings, identifying the components included there. We also illustrate each component of the class exercise in examples to show how the tool is used in the specific topic area of SV&V. Section 5 presents an example of a class exercise and how it is implemented in a classroom setting.

Section 6 describes the pedagogical evaluation strategy of the class exercises to facilitate support for our hypothesis that active, engaged learning will enhance student experience, interests and learning. Finally, Section 7 will present the summary conclusion of our initial implementation, and we invite IS educators to share access to these tools for review, use and feedback.

2. BACKGROUND AND RATIONALE

Software quality is not just a critical issue in the software industry, but it also becomes crucial in many aspects of today's information society at large. With software products being ubiquitous, it is a factor in privacy of information, in legal matters of liability, as well as national security. The fundamental challenge to a solution to improve software quality is in the people and processes that develop and produce the software products. Even after decades of development, the software industry continues to spend considerable time and effort to deal with the problem. Much of the improvement can be attributed to the implementation of standards and practices like SV&V. However survey still shows that SV&V is simply not adequately practiced in the software industry (Rakitin, 2013). Acharya et al (2014) reasoned that firstly, there is not enough awareness of the SV&V benefits, and secondly, the lack of practitioners who sufficiently understand the SV&V topics and processes.

The research project therefore aims at the root cause in the lack of SV&V courseware for effective education in academia and on-the-job training in industry. We create focus groups comprising of academic and software industry partners to critically review our existing course materials in joint effort, to identify the gaps and inadequacies when checked against current methods and best practices. Then we refine the lecture materials and develop active learning tools for teaching SV&V. We modularize the teaching materials and tools into small deliverables of 25 minutes duration and in generic formats for adaptation to various settings. The developed modules are easily integrated into software courses and can also be adapted by the industry for on-the-job training. The project's goal includes the committed support from academia and industry to sustain growth and further development for a skilled SV&V community.

Our hypothesis is that the class exercises would be effective learning tools for students since they facilitate student learning by doing and subsequently applying what they learn to solve problems in the real world. These focused exercises would enhance the understanding of the underlying theoretical concepts presented in class (and in preparatory reading) and provide a context for their application. Several class exercises developed in this work will allow the students to gain insight into the entire lifecycle of software testing process including planning, designing, implementing, recording, reporting and managing aspects of the process.

3. TOPICS AND LEARNING TOOLS

The course enhancement effort is guided by the following four specific SV&V topic areas.

- Requirements Engineering,
- Software Testing,
- Software Reviews, and
- Configuration Management.

We identified these as the critical areas in the software engineering process and areas of importance in the industry as well. The SV&V course modules are therefore based on each of these topics.

For each of these SV&V topic areas, we develop active learning tools to be used in the course modules. These learning tools include the following:

Case Studies
Case studies are drawn from industry SV&V practices. Students are presented industry standard documents for review to prepare for the tasks. These tasks may be resolution of review conflicts in the Software Requirements Specification (SRS) document, or compliance to security standards, or drafting of testing plans from use cases, but certainly are not limited to these. A more extensive coverage of the study cases developed is being disseminated in other publications (Manohar et al. 2015).

Case Study Videos
Often produced from the scripts first drafted by our industry partners and confirmed by the testimonies shared in focus group discussions, case study videos provide a realistic picture for the audience to appreciate many SV&V processes in practice. These may show how peer code review is done, and how potential tension or conflict may arise, or the tedious detailed nature of requirements solicitation.

Class Exercises
Based on the context of the class module, class exercises are designed for the class time to explicitly raise questions to invite student participation. It may be questions to think further into the concepts for a deeper understanding, or practice using their knowledge with hands-on practice for problem solving. There are many ways of using class exercises. For a small class, the teacher may simply use the exercise to engage the students in discussion and practice. For larger classes, the students can form small groups to use the class exercise as instrument leading to group projects.

Woods and Howard (2014) effectively used class exercises for Information Technology students to study ethical issues. Day and Foley (2006) used class time exclusively for exercises, having their students prepare themselves with materials provided online. Bishop and Verleger (2013) presented a comprehensive survey of the research in different ways of using class exercises, often referred to as the “flipped” classroom.

The research project is on-going, but the initial implementation of the learning modules is ready for sharing and review. The following sections will describe in further details the class exercises for the SV&V course.

4. COMPONENTS OF A CLASS EXERCISE

Each class exercise consists of the following components:

a) Exercise Description,
b) Instruction Notes,
c) Student Handout,
d) Assessment Instrument.

The Exercise Description provides the general information about the exercise. That includes the module name, the focus topic area, any prerequisite knowledge, ABET learning outcomes, keywords, expected delivery duration, and a single sentence description of what the student is supposed to do in the exercise. Figure 1 illustrates the Exercise Description in a template, filled out as Class Exercise for distinguishing between business requirements and functional requirements.

The Instruction Notes describes for the teacher how to deliver the exercise in class. It may serve as a guide card for the teacher about this exercise, but it includes materials the teacher may need to use for the exercise, such as a slide set presentation. It may also serve as a check list, a reminder about what to do, such as administering the assessment instrument at the conclusion of the class. Figure 2 illustrates that for the exercise in Requirements Engineering.

Student Handout includes everything that the students need to participate in the class exercise. Quite often it is the work sheet for the students. But it may also include other documents or artifacts for review, or tools for use. The instructor will need to prepare sufficient number for the students to use or share. In this example of Business Requirements versus Functional

Requirements exercise, the Student Handout is a work sheet, illustrated in Figure 3.

Exercise Description

Module Number: RM03

Focus Area	Requirements Engineering
Exercise Module Name	**Business Requirements versus Functional Requirements**
Prerequisite Knowledge	Before attempting this exercise module, the student should have knowledge of these terms: (a) Requirements (b) Business Requirements (c) Functional Requirements
Learning Outcomes	Upon completion of this module, the student should be able to meet the following ABET Criterion 3 outcomes. • #7- ability to communicate effectively. • #11 – ability to use the techniques, skills, and tools necessary for engineering practice.
Keywords	Requirements, Business, Functional
Duration	25 minutes x1
Exercise	For each requirement statement, identify whether it is a functional requirement or a business requirement.

Figure 1. Exercise Description

Instruction Notes

Module Number: RM03

Focus Area	Requirements Engineering
Exercise Module Name	**Business Requirements versus Functional Requirements**
Instruction	1. Print the exercise sheet and ask the students to: a. Identify the functional requirements and business requirements in the list. b. (15 minutes) 2. Use the accompanying slide set to discuss the results in class.
Assessment	Ask the students to take the survey by the assessment instrument.

Figure 2. Instruction Notes

The Assessment Instrument is a simple survey primarily for indirect assessment of student learning outcome, and also for student feedback. It is designed for generic use in every exercise, to be completed quickly at the conclusion of the class exercise. Figure 4 below is the assessment instrument for the exercise.

Software Verification & Validation Exercise

Module Number: RM03

Consider each of the following requirements statement, and write down **BR** for Business Requirement or **FR** for Functional Requirement.

BR/FR		Business Requirements vs Functional Requirements
	1	The solution will automatically validate patients against the Hospital Management System. _____
	2	The solution will enable doctors to record patient diagnosis. _____
	3	The solution will enable lab reports to be automatically sent to the respective doctor. _____
	4	We need to implement a web and mobile based employee tracking system that tracks medical staff and increases efficiency by means of monitoring medical staff activity, absenteeism and productivity. _____
	5	The system shall display the longitude and latitude of the medical staff through GPS.
	6	The system shall display the position of the medical staff on Google map.
	7	The system shall allow medical supervisors to send notifications to their medical staff.
	8	We need to establish an online patient portal.
	9	The portal should list our hospital services.
	10	The system shall be able to register a patient using the following fields: Name (20 characters max), Address (200 characters max), and Social Security Number (9 characters long).

Figure 3. Student Handout

Software Verification & Validation
Exercise Module Assessment

Module Number: RM03

Please fill out the survey.

		Yes	No
1	My team and I understood the purpose of this activity.	☐	☐
2	My team and I could complete the activity with the directions provided.	☐	☐
3	At least one member of my team was uncertain of how to carry out the steps of the activity.	☐	☐
4	The activity used a real-world application.	☐	☐
5	I could imagine carrying out this activity as part of my job.	☐	☐
6	I communicated verbally in a small group while completing this activity.	☐	☐
7	I communicated verbally in a large group while completing this activity.	☐	☐
8	I / We provided written communication as part of this activity.	☐	☐
9	I / We made a formal presentation as part of this activity.	☐	☐
10	I thought critically about the content while completing this activity.	☐	☐

Provide your overall thought about this activity.

Figure 4. Assessment Instrument

Our initial implementation at the time of this writing has completed 16 exercises, of total expected delivery time at 800 minutes. Table 1 below lists the module names each with its time duration, categorized in the four SV&V focus areas, and the category of Additional Topics in extra.

focus area	CLASS EXERCISE	min
Requirements Engineering	Ambiguous Questions	25
	Business Requirements and Functional Requirements	25
	Clarifying User Requirements	25
	Needs statements to SRS	25
	Needs statements to User Requirements	25
	Requirements Ambiguity	50
	Stated and Implied Requirements	25
Software Testing	Cost Effective Testing Approach	50
	Test Cases for a Given Requirement	50
	Testing Tools	75
	Understanding Testing	75
Software Reviews	Code Inspection	175
	SRS Review	50
Configuration Management	Defect Lifecycle	25
Additional Topics	Deming's 14 Points on System of Profound Knowledge (SoPK)	50
	Understanding IEEE Standards	50
	TOTAL	**800**

Table 1. Class Exercises

We have used one class exercise module in Requirements Engineering to illustrate the use of the class exercise components. Most of the class exercises listed here have been reviewed in focus groups. In August, 2015, a one-and-a-half-day workshop was organized for project partners and other invited participants to jointly learn and review all the learning tools we have developed: Case Studies, Case Study Videos, and Class Exercises. Eleven academic institutions and industry partners attended this workshop where we shared these tolls and discussed delivery strategies. Since the project is funded by the National Science Foundation, these course materials will be made available at the National Science Digital Library's (NSDL) national portal.

5. AN EXAMPLE CLASS EXERCISE

In this section, we use an example to show how class exercise can illustrate the problems that may arise in SV&V. Refer to Table 1 for the class exercises in the various focus areas. We take Ambiguous Questions as a class exercise example, about the gathering of requirements for an information system development project. Requirements are of two types: functional requirements and non-functional requirements (Suri & Gassert 2005). Functional requirements relate to the actions that a software product must carry out to satisfy the fundamental reasons for its existence. Non-functional requirements are the desirable properties or qualities that the software product must have for customer satisfaction. These are the characteristics pertaining to making the software product fast, usable, portable, reliable, attractive, and the like. However, it is in this area that a lot of ambiguity can arise. In class, the students are taught the terms that potentially may cause ambiguity and confusion in the requirements gathering process. The following lists some of these terms.

- acceptable, adequate.
- high-quality, state-of-the art performance.
- to the extent practicable.
- efficient.
- use-friendly.
- simple, easy, flexible.
- robust.
- seamless.
- optimal, maximal, minimal, reasonable.
- including but not limited to.
- and so on.

The class exercise will then test the students' understanding of these ambiguous terms in a requirements elicitation and gathering process. The students are given a few statements and they are asked to identify the ambiguous term or terms that sound ambiguous and discuss how they may rephrase the requirements that will clarify the meaning, making it unambiguous. The statements are:

1. For a web-based system it is required that loading of all webpages must be completed within a reasonable amount of time.

2. Access right to data is limited to the individuals with managerial rank but those with access rights may also grant access rights to others.

3. Even though the (stock) market is open during business hours, access to stock prices should be available 24/7, supporting client access to the market at the client's time locale.

4. A user should be able to customize the system behavior to cater to his/her own needs. Yet the system should provide a default case for everyone.

5. Every book is identified by the ISBN in the catalog. When a member of the library takes a book out on loan, the system must also identify which copy of the book was loaned

out, so that the member will be responsible for any damage to that specific copy of the book upon its return.

The instructor will then lead the discussion with questions to drill down the ambiguous terms identified by the students. Questions such as: Are the requirements unambiguous for the developers? Are they clear enough for the customer? Who are the customers? Do we charge the customer more if the customer cannot provide us more details? Are the requirement statements testable – so that one can demonstration satisfaction of these requirements? When should we stop discussing requirements, and become ready to do design and development? These questions can be discussed in detail.

6. PEDAGOGICAL EVALUATION

Since the class exercises are designed with specific learning objectives, we mapped the objectives to the learning outcomes derived by ABET (Accreditation Board for Engineering and Technology, Inc.) for engineering program accreditation (ABET-EAC 2014). While experienced teachers may intuitively know that good class exercises presented in an interesting way will invite student engagement into active learning, we proceeded to analyze how the ABET outcomes correspond to the levels in Bloom's taxonomy of knowledge and learning (Bloom et al 1956), to present the case of the class exercises as active learning tools, in our case, specifically for SV&V in the undergraduate curriculum. Table 2 below lists the eleven pedagogical outcomes derived by ABET pertaining to the accreditation of undergraduate engineering curriculum.

We examined the specific learning objectives of the class exercises we developed for the project. For each exercise, we identified the outcomes specifically addressed by the learning objectives as well as the other outcomes the exercise may involve but not specifically address. Figure 5 presents our results: S indicates an outcome specifically addressed, and indicates an outcome that may be involved in the exercise.

The 11 Learning Outcomes by ABET-EAC	
1	An ability to apply knowledge of mathematics, science and engineering.
2	An ability to design and conduct experiments, as well as to analyze and interpret data.
3	An ability to design a system, component or process to meet desired needs.
4	An ability to function on multidisciplinary teams.
5	An ability to identify, formulate and solve engineering problems.
6	An understanding of professional and ethical responsibilities.
7	An ability to communicate effectively.
8	A broad education necessary to understand the impact of engineering solutions in a global and societal context.
9	Recognition of the need for, and an ability to engage in lifelong learning.
10	The knowledge of contemporary issues.
11	An ability to use the techniques, skills and modern engineering tools necessary for engineering practice.

Table 2. The Learning Outcomes by ABET-EAC

Class Exercise	ABET-EAC Learning Outcomes										
	1	2	3	4	5	6	7	8	9	10	11
Ambiguous Questions				S			S				
Business vs Functional Requirements				S	I		S				
Clarifying User Requirements	S			S			S				
Needs Statements to SRS	S			S			S				I
Needs Statements to User Requirements	I			S	I	I	S	I			
Requirements Ambiguity				S			S				
Stated and Implied Requirements	I			S		I	S	I			
Cost Effective Testing Approach	I	S	I	I	I	I	S				
Test Cases for a Given Requirement	I	S		I		I	S				
Testing Tools	I	S									S
Understanding Testing	I	S	S		I		S				
Code Inspection				S		S	S				S
SRS Review	S			S		S	S				S
Defect Lifecycle					S		S				S
Deming's 14 points on SoPK							S				
Understanding IEEE standards							S				S

Figure 5. Class Exercises mapped to ABET-EAC Learning outcomes

To make the case for these exercises to be active learning tools, we adopted the revised Bloom's taxonomy for STEM (Science, Technology, Engineering and Mathematics) disciplines proposed by Girgis (2010). Table 3 lists the seven levels derived from the revised Bloom's taxonomy. We adopted the term "taxa" for each level as proposed, with the description wording specific for STEM education.

Taxa	Description
I	**Pre-Knowledge Conceptual Experiences** hands-on laboratory experiences via demonstrations, physical models, practical applications to demonstrate, visualize and observe basic concepts.
II	**Basic Conceptual Knowledge** learning, understanding, memorizing basic engineering concepts, definitions, terms, symbols, theories, laws and equations.
III	**Applied Conceptual Knowledge** solving simple concept-based problems and conducting related laboratory experiments.
IV	**Procedural Knowledge** working knowledge of solving multi-concept engineering problems.
V	**Advanced Knowledge and Analytical Skills** inter-domain and open-ended problem solving skills.
VI	**Project-based Knowledge** creative, conceptual, analytical, design, manufacturing and management skills.
VII	**Project-based Knowledge** creative, conceptual, analytical, design, manufacturing and management skills.

Table 3. Engineering Knowledge Taxonomy

We classified the ABET-EAC learning outcomes by the proposed taxonomy to present the mapping in Figure 6 below to indicate the expected "taxa" levels each outcome focuses on.

Taxa	Description	1	2	3	4	5	6	7	8	9	10	11
I	Pre-knowledge Conceptual Experience	X	X									
II	Basic Conceptual Knowledge	X	X			X			X			
III	Applied Conceptual Knowledge	X	X		X			X	X			
IV	Procedural Knowledge			X	X	X		X				
V	Adv Knowledge and Analytical Skills			X	X	X	X	X		X	X	
VI	Project-based Knowledge			X					X		X	X
VII	Professional Knowledge and Practices						X			X		X

Figure 6. ABET Outcomes classified in taxonomy

By the ABET-EAC learning outcomes each of the class exercise specifically addresses, we then determine the "taxa" levels each class exercise focuses on. Table 4 is an extension of Figure 5, to include the classification results, showing that the class exercises correspond to all the levels in the knowledge taxonomy.

The class exercises, along with other active learning tools for the project, have been readily shared since Fall 2015 to begin implementation and delivery at the partnering institutions.

CLASS EXERCISE	ABET-EAC outcomes by learning objectives	(Taxa) Knowledge Levels
Ambiguous Questions	4,7	III,IV,V
Business and Functional Requirements	4,7	III,IV,V
Clarifying User Requirements	1,4,7	I,II,III,IV,V
Needs statements to SRS	1,4,7	I,II,III,IV,V
Needs statements to User Requirements	4,7	III,IV,V
Requirements Ambiguity	4,7	III,IV,V
Stated and Implied Requirements	4,7	III,IV,V
Cost Effective Testing Approach	2,7	I,II,III,IV,V
Test Cases for a Given Requirement	2,7	I,II,III,IV,V
Testing Tools	2,11	I,II,III,VI, VII
Understanding Testing	2,3,7	I,II,III,IV, V,VI
Code Inspection	4,6,7,11	III,IV,V,VI, VII
SRS Review	1,4,6,7,11	I,II,III,IV, V,VI,VII
Defect Lifecycle	5,7,11	III,IV,V,VI, VII
Deming’s 14 Points on SoPK	7	III,IV,V
Understanding IEEE Standards	7,11	III,IV,V,VI, VII

Table 4. Classed Exercises mapped to ABET-EAC outcomes and Knowledge Taxonomy

7. SUMMARY

In a project aimed at developing a sustained community skilled in Software Verification and Validation (SV&V) for the software industry, we have embarked on critically reviewing and re-developing an undergraduate SV&V course in the software engineering curriculum. Apart from refining the lecture materials, we are using active learning tools in the flipped classroom approach. These active learning tools include Case Studies, Case Study Videos, and Class Exercises. In this paper, we reported the details of the Class Exercise in our design. The Exercise consists of four generic components: Exercise Description, Instruction Notes, Student Handout and Assessment Instrument. We illustrated the Class Exercise in one module on distinguishing between Business Requirements and Functional Requirements. The finished Class Exercises in our initial implementation are also reported. To evaluate them, we further mapped the learning objectives of the Class Exercises to the expected learning outcomes for ABET accreditation of an undergraduate engineering program. We also analyzed them based on the classification by Bloom's taxonomy of knowledge adapted to software engineering. The finished Class Exercises in our initial implementation are also reported. The Class Exercises, along with other learning tools developed in the project are being reviewed and will subsequently be shared publicly in the NSDL portal.

8. ACKNOWLEDGEMENT

The authors acknowledge the support of NSF through an NSF-TUES grant entitled "Collaborative Education: Building a Skilled V&V Community", Award # 1245036. The input of industrial partners for this project including Eaton Electrical Corporation, PNC Bank, ANSYS, ServiceLink, and JDA Software Group is gratefully acknowledged. The support of the academic development partners Virginia State University (an HBCU), and Milwaukee School of Engineering is highly appreciated. In addition input from academic implementation partners including Embry-Riddle Aeronautical University, Montana Technical University, University of Michigan-Dearborn, Fairfield University, Auburn University, East Carolina University, Kennesaw State University, Bowie State University, and Clarion University is appreciated.

9. REFERENCES

ABET-EAC. (2014). Criteria for Accrediting Engineering Programs (2014-2015 Cycle). Accreditation Board for Engineering and Technology (ABET) Engineering Accreditation Commission (EAC), published March 2014. http://www.abet.org/wp-content/uploads/2015/04/E001-14-15-EAC-Criteria.pdf

Acharya, Sushil, Manohar P., Schilling Jr W.W., Ansari A.A., & Wu P.Y. (2014) Collaborative Education: Building a Skilled Software Verification and Validation User Community. 121st ASEE Annual Conference & Exposition, June 2014, Indianapolis, IN.

Bertha, C. (2010). How to teach engineering ethics course with case studies. Proceedings of the 117th ASEE Annual Conference and Exposition, Louisville, KY.

Bishop, J.L. & Verleger M.A. (2013). The Flipped Classroom: A Survey of the Research. ASEE 120th Annual Conference and Exposition, Atlanta, GA.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. David McKay Company, New York.

Day, J.A. & Foley J.D. (2006) Evaluating a Web Lecture Intervention in a Human-Computer Interaction Course. IEEE Transactions on Education 49(4):420-431, 2006.

Frydenberg, Mark. (2013). Flipping Excel. Information Systems Education Journal, 11(1), pp.63-73. http://isedj.org/2013-11/ ISSN: 1545-679X.

Girgis, M. (2010) A New Engineering Taxonomy for Assessing Conceptual and Problem-Solving Competencies. 117th ASEE Annual Conference & Exposition, Louisville, KY.

Lopez, Gustavo, Cocozza F., Martinez A. and Jenkins M. (2015). Design and Implementation of a Software Testing Training Course. 122*nd* ASEE Annual Conference & Exposition, June 2015, Seattle, WA.

Manohar, Priya, Acharya S., Wu P.Y., Ansari A.A. & Schilling Jr W.W. (2015). Case Study Based Educational Tools for Teaching Software V&V Course at Undergraduate Level. 122*nd* ASEE Annual Conference & Exposition, June 2015, Seattle, WA.

Rakitin, Steven R. (2000). Software Verification and Validation for Practitioners and Managers by Steve R. Rakitin, 2nd Edition, Artech House. ISBN: 1-58053-296-9.

Strayer, Jeremy F. (2012). How Learning in an Inverted Classroom Influences Cooperation, Innovation and Task Orientation. Learning Environments Research 15(2):171-193.

Suri, Deepti and Gassert J. (2005). Gathering project Requirements: A Collaborative and Interdisciplinary Experience. 112*th* ASEE

Annual Conference & Exposition, June 2005, Portland, OR.

Woods, D. & Howard, E. (2014) An Active Learning Activity for an IT Ethics Course. Information System Education Journal 12(1):73-77. http://isedj.org/2014-12/ISSN:1545-679X.

The Application of Writing Across the Curriculum (WAC) Techniques in a Systems Analysis & Design Flipped Classroom

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

Abstract

To more effectively meet the expectations of industry for entry-level IT employees, a case is made for the inclusion of writing throughout the Computer Information Systems (CIS) curriculum. *Writing Across the Curriculum (WAC)* principles are explained, and it is opined that both Writing to Learn (WTL) and Writing in the Disciplines (WID) pedagogies are desirable for inclusion as part of the delivery of the CIS curriculum. Examples of both WTL and WID are provided from the author's Systems Analysis & Design (SAD) course. It is concluded that the use of WTL and WID techniques in the flipped SAD course has both (1) increase student engagement both in and out of the classroom, and (2) improved student writing and learning.

Keywords: Student Learning, Student Engagement, Employer Expectations, Writing Across the Curriculum, Writing in the Discipline, Writing to Learn.

1. WRITING IN THE CIS CURRICULUM?

We live in an era of sound bites and 140 character messages, but good writing is still necessary for success in today's business environment. Paying attention to grammar, spelling and punctuation, along with good word choice and the use of a consistent style, is important because bad writing can have a wide range of negative career and personal consequences.

As information systems professionals we use writing daily for a variety of purposes including to communicate information (memos, email, etc.), to clarify our thinking (when we work through an idea or problem in writing), to learn new concepts and information (taking notes on reading and research topics), and to write formal reports (requirements definition, feasibility study, systems proposals, etc.).

As aspiring professionals our students need practice to be able to use writing effectively to meet these same goals. One or two writing classes taken in the freshman year simply cannot provide enough practice to increase the quality of our students thinking and writing. As one response to students' lack of writing practice throughout the university curriculum, *Writing Across the Curriculum* (WAC) programs began to emerge in the early 1980s. While the structure of individual WAC programs exhibit some degree of variation, the philosophies underlying these programs generally agree on certain basic principles: (1) writing is the responsibility of the entire academic community; (2) writing must be integrated across departmental boundaries; (3) writing must be continuous during all four years of undergraduate education; (4) writing promotes learning; and (5) only by practicing the conventions of an academic discipline will students begin to communicate effectively within that discipline.

Many recent studies of employer expectations of information systems graduates (Hart, 2013) (Pratt, Keys & Wirkus, 2014) have reaffirmed the need to focus on improving the writing skills of our graduates. Curriculum requirements of professional organizations (ABET, 2013) (AAC&U, 2013) (AACSB, 2013) (Attaway, Chandra, Dos Santos, Thatcher & Wright, 2011) (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, & De Vreeda, 2010) have reached the same conclusions.

Including writing in courses has both short- and long-term benefits for both teachers and students. In the short term, teachers are better able to gauge how well students grasp information and where they need elaboration of key concepts. In the long run, as more Computer Information Systems (CIS) professors incorporate writing into more courses, students become more efficient at using writing as a communication and learning tool. Especially for more advanced or specialized work in the discipline, the professors reap the benefits of having students who are much better grounded in the fundamentals and ready to engage in more sophisticated analysis of ideas.

Like all language skills, writing skills atrophy when they aren't used. Yet our students often report that they do *no writing at all* during a semester because they don't even take notes during some classes. For students who take only multiple-choice exams, writing can be avoided almost completely for months at a time. Assigning writing in all courses helps students keep their writing skills sharp. Moreover, faculty in all disciplines have discovered that assigning writing in their classes helps students learn material and improve their thinking about ideas in the courses. Writing assigned across the curriculum also helps students prepare for the day-in and day-out communication tasks they'll face on the job, no matter what their job is. Equally important, students need to learn about how writing is used within a discipline, and many kinds of assignments give students practice with disciplinary forms and conventions.

So why assign writing in your Information Systems classes? Students will learn more and will leave the university better prepared to face communication challenges of the profession if they write consistently over the course of a four-year college program. Additionally and much more specifically, students will learn more about the material in their courses at a much greater depth if professors assign writing for their courses.

2. WRITING TO LEARN (WTL)

When considering how Writing across the Curriculum (WAC) has been implemented at a range of universities, the writing assignments generally fall into one of two categories – Writing to Learn (WTL) and Writing in the Disciplines (WID). While some teachers combine the two categories and assign writing that meets the goals of each, many teachers choose to focus on one type or the other.

Writing-to-Learn (WTL) activities are short, impromptu or otherwise informal writing tasks that help students think through and/or discover key concepts or ideas presented in a course (Forsman, 1985). Often, these writing tasks are limited to less than five minutes of class time or are assigned as brief, out-of-class assignments. Writing-to-learn (WTL) activities are considered to be crucial by many WAC programs because they can be used as evidence that students have learned the information and/or suggest areas in which there is an information deficit.

Writing to Learn (WTL) activities can happen frequently or infrequently in a typical class setting. Some can extend over the entire semester, whereas others can be extended to include a wide variety of writing tasks in different formats and to different audiences. Because they are examples of informal writing and are often given impromptu, WTL activities usually aren't marked for correctness. Rather, teachers or classmates quickly read the writing for a general sense of what students understand and don't understand. These activities take very little class time, and most teachers find they can give a quick WTL prompt at the beginning of class while they take roll and as students are settling in. Moreover, many WTL activities can be limited to just a minute or two--the amount of time it might take to answer a student's question about a course concept. Also, because WTL activities are such valuable learning tools, most teachers feel that student’s use of any minutes given over to WTL writing is a very effective use of class time.

Although it is not clear exactly how writing fosters critical thinking (Applebee, 1985), both theoreticians and practitioners agree that writing promotes both critical thinking and improved learning (Adams, 1972) (Bruner, 1975) (Emig, 1977) (Herrington, 1981) (Knoblauch & Brannon,

1983) (Odell, 1980) (Parker & Goodkin, 1987). As Fulwiler and Young so succinctly put it (1982, p. x), "Writing to communicate--or what James Britton calls "transactional writing"--means writing to accomplish something, to inform, instruct, or persuade. Writing to learn is different. We write to ourselves as well as talk with others to objectify our perceptions of reality; the primary function of this "expressive" language is not to communicate, but to order and represent experience to our own understanding. In this sense language provides us with a unique way of knowing and becomes a tool for discovering, for shaping meaning, and for reaching understanding."

Forsman (1985, p.9) makes the same point, but she directs her attention not to a theoretical justification but to a practical rationale for writing to learn. Forsman states, "As teachers we can choose between (a) sentencing students to thoughtless mechanical operations and (b) facilitating their ability to think. If students' readiness for more involved thought processes is bypassed in favor of jamming more facts and figures into their heads, they will stagnate at the lower levels of thinking. But if students are encouraged to try a variety of thought processes in classes, they can, regardless of their ages, develop considerable mental power. Writing is one of the most effective ways to develop thinking."

3. WRITING TO LEARN IN THE SYSTEMS ANALYSIS & DESIGN COURSE

WTL in the Flipped Classroom Approach

Writing to Learn prompts are used in the CIS curriculum as part of the author's "flipped classroom" approach to the Systems Analysis & Design (SAD) course. While different authors espouse different key components of the flipped classroom, there are several components that are essential to all interpretations of the flipped classroom including the following.

The Flipped Classroom approach provides an Opportunity for Students to gain First Exposure to Content Prior to Class. The mechanism used for first exposure can vary, from simple textbook or online readings to lecture videos to podcasts or screencasts. Videos can be created by the course instructor, or found online from sources such as YouTube, the Kahn Academy, MIT's OpenCourseWare, or other similar sources. The pre-class exposure does not need to be high-tech; students can be asked to simply complete pre-class reading assignments and/or engage in writing-to-learn exercises.

Use of the Flipped Classroom should provide an Incentive for Students to Prepare for Class. In all cases, students should be required to complete a task associated with their preparation, and that task should be associated with some points or percentage toward their final course grade. The assignments themselves can vary, ranging from online quizzes to worksheets to short writing-to-learn assignments. In each case the task should provide an incentive for students to come to class prepared by speaking the common language of undergraduates: points. In many cases grading for completion rather than effort may be sufficient, particularly if in-class activities will provide students with the kind of feedback that grading for accuracy usually provides.

The Pre-Class Activity Should Provide a Mechanism to Assess Student Understanding. The pre-class assignments that students complete as evidence of their preparation can also help both the instructor and the student assess understanding. Pre-class quizzes can also allow the instructor to practice Just-in-Time Teaching (Novak, Patterson, Gavrin, & Christian, 1999), which means that the instructor can tailor class activities to focus on the elements with which students are struggling. If automatically graded, the quizzes can also help students pinpoint areas where they need help. Pre-class worksheets also can help focus student attention on areas with which they are struggling, and can serve as a departure point for in-class activities, while pre-class writing assignments can help students clarify their thinking about a subject, thereby providing for richer in-class discussions. Most importantly, the use of pre-class activities provides for the time needed to supply students with much needed feedback in class, reducing the need for instructors to provide extensive commentary outside of class (Walvoord, 1992). Additionally, many of the activities used during class time (e.g., clicker questions, debates, etc.) can serve as informal checks of student learning.

Use of the Flipped Classroom Should Provide In-Class Activities that Focus on Higher Level Cognitive Activities. Given that the students have gained basic knowledge outside of class, class time can now be spent promoting deeper learning. The in-class activity selected will be dependent upon both the learning goals of the course and the culture of the discipline. For example, Lage (2000) describes experiments students did in class to illustrate economic principles, while Mazur (2009) focuses on student discussion of conceptual "clicker" questions and quantitative problems which focused on physical science principles. Other in class activities may consist of debates, data analysis, or synthesis activities. What is important, regardless of the activity chosen, is that students are using class time to deepen their understanding and increase their skills at using their newly acquired knowledge.

Writing to Learn in the Flipped (SAD) Course
The use of WTL prompts as part of the flipped classroom approach for the SAD course was first implemented in the fall of 2013, and is currently in its third iteration. The course consists of an opening unit which concludes with students being assigned to project teams, after which the teams are assigned to improve a particular system/application in either the on campus or off campus environments.

The typical daily classroom unit consists of the following pattern: (1) Students receive a Research Question (WTL prompt) at the close of the prior class; (2) Students post their individual answers to the research question to the Blackboard course management system no later than midnight of the evening prior to the class session in which the topic will be covered in class; (3) The actual class session opens with either an additional WTL prompt followed by a 10-15 minute comparison of the student answers, or just the 10-15 minute comparison of student answers. During the 10-15 minute session students search for commonalities in their answers, following which the students collectively decide which information to archive for summative assessment at a later date; and (4) Students then apply their understanding of the answers to their particular system or application.

For example, applying the pattern to the SAD class session covering System Requirements:

- Student Research Question: What are system requirements? What is the difference between functional system requirements and non-functional system requirements?
- In class, after the opening discussion, student project teams work together to define the functional and non-functional system requirements for their system or application.
- Project teams quickly come to realize that they cannot accurately define their requirements without input from the system stakeholders (a topic that was covered 2 weeks earlier in the course).
- Students are then provided with their research question and/or assignment that is due prior to the next class session. In particular, students are asked to find commonly employed techniques to gather data and to determine which data gathering technique(s) would be most appropriate to collect data from each class of stakeholder, which provides input for the following class session which covers Data Gathering Techniques.

Student postings to the course management system are usually graded on a 2-point scale with 0 = answer not submitted by the deadline; 1 = standard Wikipedia answer; 2 = additional source(s) used to provide their answer. No late postings are accepted because all research assignments are posted to the course management system well in advance of the required due dates.

At the close of each class session, students and faculty together decide which information would be best archived for those students who missed class due to illness or other higher priority interventions. This archived material becomes the basis for summative assessments that provide for individual accountability in what is predominantly a team-based course producing team-based project deliverables.

The flipped classroom approach employed herein is not merely a synonym for either viewing online videos or searching for information on the internet. Rather, it is the in-class interactions and carefully designed learning activities that occur during classroom face-to-face time that are the most important part of the flipped classroom. Students do not work without structure in class, nor do they work in isolation, nor do they spend the entire class time staring at a computer screen. Rather, the flipped classroom is an opportunity to increase intentionally designed and meaningful interactions between students and faculty.

4. WRITING IN THE DISCIPLINES (WID)

Writing in the Disciplines
The second category of WAC is called Writing in the Disciplines (WID). WID assignments are designed to introduce or give students practice with both the language conventions and specific formats typical of a given discipline. For example,

a feasibility study would include much different information in a unique format that would differ greatly from an annual business report or an engineering lab report.
Most WID assignments are formal documents prepared over a few weeks or even months. The final documents adhere to the format and style guidelines of project deliverables typical of the professional genres they are helping students learn about. Teachers comment primarily on the substance of these assignments, but teachers also expect students to meet professional standards of both layout and proofreading (Russell, 1991).

Although the research essay is the most common kind of WID assignment, there are many other formats that can be used to teach students about disciplinary writing conventions. For example, in the Systems Analysis & Design course students produce the following types of documents to understand the thinking and writing of the systems development process: (1) a record of the initial client meeting(s); (2) a systems requirements document; (3) a feasibility study; etc.

In addition to discipline-specific formats, other kinds of writing assignments can help students learn the language and ways of thinking of a discipline, even though they may not mimic its professional writing. Any of the following writing activities can provide the basis for a longer, more formal assignment, or can be used only to promote class discussion and/or thinking about course material: (1) *Reading Journa*l – jotting down specific thoughts regarding an assigned reading; (2) *Jargon/Term Journal* – jotting down terms and their meanings when confronted with new discipline specific terminology; (3) *Rhetorical Analysis* – reading an article and analyzing its meaning; for example, a mini-case study emphasizing an important point; and (4) *Popular Article* – for example, our students do a "Tech Week in Review" single-page analysis of tech happenings on a weekly basis in our introductory course

One reason that students report feeling overwhelmed by WID tasks is that they aren't sure where to start and then how to proceed to produce a good project of the sort required by the assignment. One can assist students--and consequently receive better final drafts to read--by setting up a sequence of tasks that build toward the final project. Two approaches work well when designing a sequence: (1) break the large writing task into chunks so that students can tackle parts of the assignment and get feedback before moving to the next chunk; or (2) alternately, devise tasks that build on each other. For instance, if the assignment is to conduct a professional literature review as the final project, first have students write abstracts or summaries of articles, then ask for annotations, and finally ask for synthesis. At the same time, have students analyze published articles to determine what a review of literature typically looks like in the field. By giving students a sequence of writing and analytic tasks, they become more confident and more able to meet the criteria for the final writing task.

Successful writing assignments depend on careful and thorough instructions and preparation and on explicit criteria for evaluation. Although individual experience with a given assignment will suggest ways for assignment improvement, following explicit guidelines in initial assignment construction can assist in avoiding potential problems with student writing and thus makes for both better writing and thereby considerably reduced grading time. Good writing assignments always start with a clear goal that the teacher can express, usually included on the assignment sheet so that students also understand the goal of the assignment.

Good writing assignments also often take shape by thinking backwards; in effect, teachers ask themselves, "What do I want to read at the end of this assignment?" By working from what they anticipate the final product should look like, teachers can give students detailed guidelines about both the writing task and the final written product, including: (1) rhetorical aspects of the task; i.e., who is the audience for the finished product, the purpose of the assignment, and writing situation; (2) required sections and elements of the writing assignment; (3) grading criteria which should be included on the assignment sheet; and then (4) breaking down the writing task into a manageable steps.

5. WID IN THE SYSTEMS ANALYSIS & DESIGN COURSE

Discipline based writing assignments are present in almost all required CIS courses at the author's university. In the Systems Analysis & Design (SAD) course project teams do an analysis and redesign of a real world information system. Project teams do several WID activities commencing with the developing a Team Charter, a formal document that defines the purpose of the team, expected outcomes, and ground rules for working together to produce the results. In effect, a team charter is a set of agreements created to ensure that everyone is on the same page regarding project team norms from the start of the project.

Following the development of the team charter, project teams are assigned to one of several real-world applications and required to produce specific deliverables through the Requirements, Analysis, and Design phases of the Systems Development Life Cycle (SDLC). Specific WID assignments treated as project deliverables in the author's SAD course include: (1) a summary report of the *initial client meeting*, a meeting held to define the projects goals, define data availability, discuss implementation issues, and discuss project planning/scheduling; (2) a *problem definition statement* which is a concise description of the issues that need to be addressed by the project team; (3) a *system scope statement* which in effect establishes the boundaries of the study by establishing the project deliverables and major objectives of the project; (4) a *system request* which formally establishes the project goals and objectives ; (5) a *feasibility analysis* which is conducted to determine if the problem can be solved effectively from operational (will it work?), economic (costs and benefits), and technical (can it be built?) viewpoints; (6) a *requirements definition report* which formally establishes what the system must produce within the established organizational parameters; (7) a *system specification* (which includes the leveled set of DFD's, process specifications, and a data dictionary; and (8) a *system proposal* which includes all of the first seven components preceded by an executive summary.

6. RESULTS & CONCLUSIONS

The use of WTL prompts both prior to and/or at the beginning of class sessions has increased student engagement both outside of the classroom and inside the classroom. Students have increased their reading outside the classroom because they have read online material to answer research questions related to course content prior to encountering the content in their classes. Because they have available to them the answers that they submitted the evening prior to the class, students are much more engaged in the daily course content which serves to foster much more lively class discussions.

The use of WID assignments in successive courses has both addressed the need for effective writing and increased the quality of student writing as students have progressed through the curriculum. As the quality of student writing has improved via the constant reinforcement of writing assignments, students have learned to think at a higher level. As Zakaria (2015) so effectively proffers, "The central virtue of a Liberal Education is that it teaches you how to write, and writing makes you think. Whatever you do in life, the ability to write clearly, cleanly, and reasonably quickly will prove to be an invaluable skill."

The presence of WTL assignments which increase student engagement in the course content both in and out of the classroom, coupled with the use of WID assignments in successive courses which have greatly increased the individual student's writing ability, has effectively produced undergraduates whose writing skills and field-specific knowledge are closer to being in alignment with the expectations of the employers of our graduates. Though longitudinal studies are nearly impossible to conduct due to the absence of a control group, individual student writing has exhibited great improvement over the course of their four-year undergraduate experience.

7. REFERENCES

ABET, Computing Accreditation Commission (CAC) (2013). Criteria for Accrediting Computing Programs, 2014-2015. Retrieved April 13, 2015 from http://www.abet.org/cac-criteria-2014-2015/

Adams, P. (Ed.) (1973). *Language in Thinking*. Harmondsworth: Penguin Press.

Applebee, A.N. (1985). Writing and Reasoning. *Review of Educational Research, 54*(4), 577-596.

Association of American Colleges & Universities (AAC&U) (2013). Liberal Education and America's Promise (LEAP). Retrieved April 13, 2015 from **http://www.aacu.org/leap**

Association to Advance Collegiate Schools of Business (AACSB) (2013). Eligibility Procedures and Accreditation Standards for Business Accreditation. Retrieved April 13, 2015 from http://www.aacsb.edu/en/accreditation/standards/2013-business.aspx

Attaway, A., Chandra, S., Dos Santos, B., Thatcher, M., & Wright, A. (2011). An Approach to Meeting AACSB Assurance of Learning Standards in an IS Core Course. *Journal of Information Systems Education, 22(4),* 355-366*.*

Bruner, J. (1975). Language as an Instrument of Thought. In A. Davies (Ed.), *Problems in language and learning*. London: Heinemann.

Emig, J. (1977). Writing as a Mode of Learning. *College Composition and Communication, 28*, 122-28.

Forsman, S. (1985). Writing to Learn Means Learning to Think. In A. R. Gere (Ed.), *Roots in the sawdust: Writing to learn across the disciplines* (pp. 162-174). Urbana, IL: National Council of Teachers of English.

Fulwiler, T. & Young, A. (1982). Introduction. In T. Fulwiler and A. Young (Eds.), *Language connections: Writing and reading across the curriculum* (pp. ix-xiii). Urbana, IL: National Council of Teachers of English.

Hart Research Associates (2013). *It Takes More than a Major: Employer Priorities for College Learning and Student Success.* Washington, DC: The Association of American Colleges and Universities.

Herrington, A. (1981). Writing to Learn: Writing Across the Disciplines. *College English, 43*, 379-87.

Knoblauch, C., & Brannon, L. (1983). Writing as Learning through the Curriculum. *College English, 45*, 465-74.

Lage, M., Platt, G. & Treglia, M. (2000). Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. *Journal of Economic Education, 31*:30-43.

Mazur, E. (2009). Farewell, Lecture? *Science* 323: 50-51.

Novak, G., Patterson, E., Gavrin, A., & Christian, W. (1999). *Just-in-Time Teaching: Blending Active Learning with Web Technology.* Upper Saddle River, NJ: Prentice Hall.

Odell, L. (1980). The Process of Writing and the Process of Learning. *College Composition and Communication, 36*, 42-50.

Parker, R. P., & Goodkin, V. (1987). *The Consequences of Writing: Enhancing Learning in the Disciplines*. Upper Montclair, NJ: Boynton/Cook.

Pratt, J., Keys, A., & Wirkus, T. (2014). Preparing Information Systems Graduates for a Complex Society: Aligning IS Curricula with Liberal Education Learning Outcomes, *Journal of Information Systems Education, 25(1),* 5-44*.*

Russell, D. R. (1991*). Writing in the Academic Disciplines, 1870-1990: A Curricular History*. Carbondale: Southern Illinois University Press.

Saulnier, B. (2015). The Flipped Classroom in Systems Analysis & Design: Leveraging Technology to Increase Student Engagement. *Information Systems Education Journal, 13(4),* 33-40*.*

Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker, Jr, J, Sipior, J., & De Vreeda, G.-J. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of AIS, 26,* 359-428.

Walvoord, B.E. (1992). Getting started. In *Writing across the curriculum: A guide to developing programs*, edited by S.H. McLeod and M. Soven. Newbury Park, CA: Sage.

Walvoord, B. & Anderson, V. (1998). *Effective Grading: A Tool for Learning and Assessment.* San Francisco: Jossey-Bass.

Zakaria, F. (2015). *In Defense of a Liberal Education.* New York: W.W.Norton.

Editor's Note:

This paper was selected for inclusion in the journal as a EDSIGCon 2015 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 2015.

A Topic Analysis of ISECON Conference Proceedings from 1982 through 2014

Jon Clark
jon.clark@colostate.edu

Susan Athey
susan.athey@colostate.edu

John Plotnicki
john.plotnicki@colostate.edu

Jay Barnes
jjbarnes@rams.colostate.edu

Computer Information Systems
Colorado State University
Ft. Collins, CO 80523 USA

Abstract

The authors note a distinct shift in topics covered in curricula as well as in conference presentations. This research was undertaken to get a better understanding of what these shifts have been, and determine their magnitude over time. Since ISECON has published its conference proceedings in digital format since 1982, this was a logical source of topics on which to base the analysis. All proceedings were captured during the period of 1982 through 2014 and a series of keyword searches were undertaken based on approximately 130 topics. One significant finding is that technical topics, programming as an example, has seen a steady and substantial decline from 18% of topics to just 2.5%. The most precipitous drop occurred from 1993 through 1997. This paper discusses the various trends and hypothesizes as to the causes.

Keywords: Curricula topics, ISECON Topic Trends, Business education gap, Technical skills, Hirable IT skills.

1. INTRODUCTION

Faculty that joined academia in the 1970's have seen many changes to the core knowledge required of graduates in information systems at the bachelors, masters and even the doctoral level. Standard curricula have changed as well with the emphasis shifting from very technical skills to softer skills. The IS'97 Model included programming and data structures in the curriculum requirements but the IS2010 Model specifically removes Application Development as a requirement (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, deVreede 2010, p.27) .

The purpose of the paper is to illustrate the topical changes and the ISECON conference over the last 32 years and act as a catalyst for discussion about both Information Systems education, and the idea of the "business-education" gap that some feel exists. While the implication of the trends in topics is certainly debatable, it is important to note that the trends themselves are based directly on the data.

Clearly, some limitations and challenges exist with this approach. ISECON did not take place in 1984, 1989, 1991 and many of the conferences had a theme and would have attracted topics in keeping with the theme. During the years of 1999 to 2014, the ISECON proceedings were a joint publication of FITE (Foundation for Information Technology Education) and EDSIG/AITP (Education Special Interest Group of AITP).

2. METHODOLOGY

The ISECON digital library of conference proceedings http://proc.isecon.org/ (Foundation for Information Technology Education, 2015). The ISECON conference proceedings have been published since 1982. The conference was not held in 1984, 1989 and 1991. This research distilled over 8000 keywords from articles at conferences over a 32-year span into categories that could be quantifiably analyzed. As there is no set process for this type of data grooming, the categories were not determined in advance, but evolved and revealed themselves as the data was organized. While might argue that the categories are somewhat subjective in nature, we submit that the analysis results and trends are both usable and provocative. We would also like to point out that the terms Computer Information Systems, Management Information Systems and Computer Science (CIS MIS and CS), ALL specifically appear in the source data from the conference, and is mainly geared towards CIS/MIS, CS appears and has been increasing in discussion since the year 2000.

The goal was to obtain, sort and categorize keywords from articles published each year at ISECON. Papers from 2000 – 2014 use keywords supplied by the authors of the original papers. Keywords in other years were determined by authors of this paper by reading the papers themselves. This generated 8390 keywords.

The keyword list was inspected and "groomed" for consistency. This included changing whole word terms into acronyms when both were used (examples: AI / Artificial Intelligence, and MIS / Management Information Systems), correcting some misspelling and separating composite keywords that had two distinct meanings. The cleaned up keywords were then assigned to categories, which could have both a primary and secondary part such as EDU-Teaching for education group, teaching keyword. The categories evolved as the words were sorted. In the end there were 121 categories created.

3. SUMMARY STATISTICS

Table 1 shows the keyword and article counts by year and average keywords per article. There is a significant increase in the number of keywords per article starting in 1999.

Year	KW's	Articles	KW's / Article
1982	22	10	2.2
1983	49	14	3.5
1984	No conference this year		
1985	125	47	2.7
1986	204	74	2.8
1987	258	78	3.3
1988	146	48	3
1989	No conference this year		
1990	133	53	2.5
1991	No conference this year		
1992	101	34	3
1993	105	43	2.4
1994	109	40	2.7
1995	134	48	2.8
1996	59	26	2.3
1997	90	34	2.6
1998	142	60	2.4
1999	200	51	3.9
2000	519	118	4.4
2001	416	94	4.4
2002	338	75	4.5
2003	418	91	4.6
2004	508	105	4.8
2005	553	115	4.8
2006	545	112	4.9
2007	566	113	5
2008	403	76	5.3
2009	408	81	5
2010	456	89	5.1
2011	386	72	5.4
2012	341	66	5.2
2013	315	65	4.8
2014	341	73	4.7
Totals	8390	2005	
Ave.	254.2	60.8	4.2

Table 1: Keywords per Year

Years 1982 through 1998 show 2.2 to 3.3 keywords per article, averaging 2.5, and 1999 to 2014 show 3.9 to 5.4 keywords per article, averaging 4.9- almost double the number of keywords per article. We did not have details on article submission requirements that far back, but one possible explanation for the difference could be a change in the allowable or minimum required keywords for articles.

The keywords have been grouped into categories for analysis. The categories evolved as the data was organized and groomed. Table 2 shows the total number of unique categories for each year. As with the number of keywords, there is a related increase in categories beginning in 2000.

Table 12 in the Appendix includes the total list of categories and number of keywords in each category.

Year	Category Count		Year	Category Count
1982	13		1999	67
1983	25		2000	91
1984	N/A		2001	83
1985	40		2002	72
1986	55		2003	84
1987	57		2004	94
1988	54		2005	91
1989	N/A		2006	88
1990	56		2007	88
1991	N/A		2008	84
1992	49		2009	91
1993	49		2010	82
1994	52		2011	80
1995	62		2012	79
1996	33		2013	77
1997	41		2014	70
1998	60			

Table 2: Unique Categories per Year

Table 3 shows the top 10 categories by number of keywords in the category. Since the ISECON conference is about education, it is no surprise that the EDU-Curriculum category contained the most keywords at 734, about 9% of the total.

At this point, one might be inclined to discuss 'unique' keywords as well as 'categories'; however, due to the nature of the data, many keywords are quite similar, with virtually identical meaning. An example would be "online education" and "distance education". While these will show up as different "unique" keywords, they have a very similar, if not the same, contextual meaning. Some terms change over time, but have the same meaning and at times are used interchangeably in a given situation. This makes metrics about unique keywords potentially misleading as the numbers of keywords with different meanings could be considerably lower than the actual number of 'unique' keywords shown. The ambiguous level of granularity between keywords would prove difficult if not impossible to properly analyze. Therefore, the discussion in this paper is centered around trends found in the categories, and some generic matching of keywords that do not have this ambiguity of meaning. For example, COBOL and JAVA are both languages, but are discrete and different in meaning, so they can be matched as individual words, not just in PROG categories.

Category List	Keywords
Edu-Curriculum	734
General	561
Edu-Teaching	374
Edu-Distance Teaching	276
Edu-General	227
IT-Discipline	213
Software-Development	207
Edu-Course	200
Social-Human Factor	197
Edu-Learning	178

Table 3: Top 10 Keyword Categories

"Software development" is the only technical topic to show up in the top ten categories list. The category "IT-Discipline" contains general terms like "Computer Science" or "Information Systems" used by themselves as a keyword, with no other descriptive words. The "General" category is mostly single non-descript keywords such as "opportunity" or "methodology".

4. ANALYSIS AND OBSERVED TRENDS

This section presents a series of graphs and tables depicting the trends of various clusters of keywords from the first ISECON Conference held in 1982 through last year's conference in 2014. Three general topics are discussed with respect to the ISECON conference keyword data analysis:

The Industry / Education Gap, Specific programming and IT skill trends, and the impact of Social topics on ISECON topics.

The Industry Education Gap
The 1999 ISECON conference mission statement was "*Bringing Industry and Education together--Closing the IS resource Gap*". While the industry education gap is a common topic in both business and academic circles, how to address this issue sparks much discussion and debate, yet remains elusive. Some aspects of this are curriculum, in terms of what skills to teach in the changing technology field, how to keep up on current technology without getting lost on fad or dead end developments, and how much and how fast students can learn and still retain the knowledge and skills.

Technical, education and business keyword clusters were plotted (Figure 1) and several important observations can be made: 1) from 1982 through 1995 education topics dropped from 40% to around 27% of the proceedings; 2) technical topics saw a rather precipitous drop from a peak of 56% in 1990, to under 20% by 2014; 3) business topics have been remarkably consistent in the range of 2 to 12% over the entire 32 year period, but have remained under about 8% since 2000. (Note: All figures are included in the appendix in a larger more detailed format.)

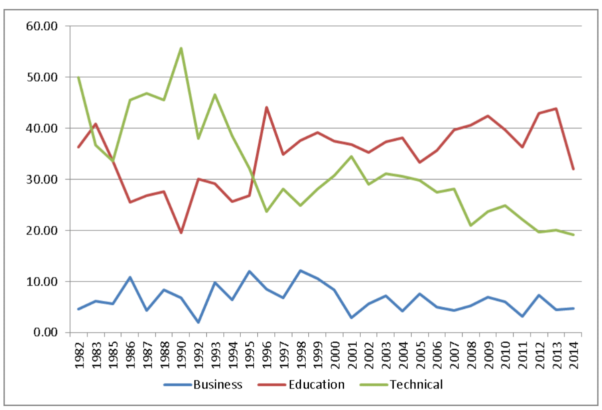


Figure 1: Technical/Education/Business (As a percent of total KW's by year)

Since ISECON is an educator's conference, it's not surprising that a large proportion of the material submitted is instructional in nature. The period from 1983 through 1995 is interesting because of the decrease in education related keywords, but this may be explained as an increase in new technical topics in the rapidly evolving environment of the computer field during this time frame. The apparent inverse relationship between education and technical terms between 1985 and 1996 deserves exploration, as well as the trends that took place from 1996 through 2014. These two seemingly distinct time frames are worth conjecture and further study.

The overall drop in technical topics at the conference since 1990 is counter intuitive. One would think that a technical field whose very origin is computer programming would not have seen such a precipitous drop of coverage in the conference. In fact, based on feedback we receive from the departmental advisory council for CIS at Colorado State University, programming is still a foundation skill for the graduates they want to hire. Are technical topics are being abandoned for educational topics in a disproportionate amount? If this is a reflection on the subjects covered in an undergraduate IS degree, this would seem to increase the Industry - Education gap by lowering the focus on skills that business wants graduates to have.

Figure 2 shows the coverage of MIS, CIS and CS as **disciplines** of the technology field over time. One trend is the increase in conference articles mentioning CS (computer science) as compared to CIS/MIS (Computer Information Systems / Management Information Systems), indicating the more technical computer science degree is discussed more, while technical topics like programming are discussed less.

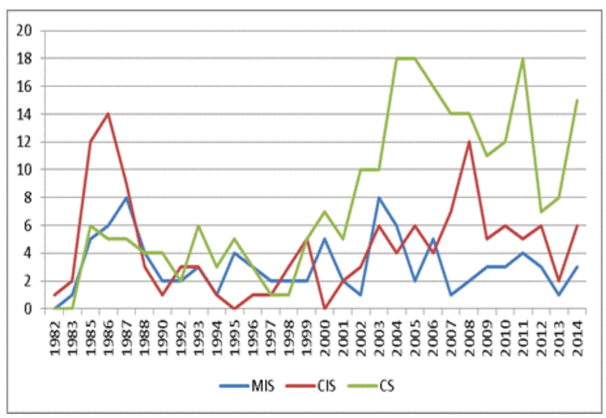


Figure 2: MIS/CIS/CS keywords (As number of keywords by year)

Figure 3 shows just two keyword clusters, programming and education, as a clear example of the gap between education and demand for technical skills. Although education topics saw an early drop from 40% in the period of 1982 through 1995, there was resurgence in publication activity on these topics following 1995 and a relatively stable 35% to 40% level of publication coverage since. Programming, however, began at 18% in 1982 and has stumbled along after dropping throughout the 1990's to under 5% since 2008.

This widening gap clearly shows that topics at ISECON have moved from more technical topics in IS education, in this case, programming, to Education in IS. The data seems to show that the conference has shifted from "what technologies to teach", to "how to use technology to teach".

Figure 4 substantiates the previous assertion. It shows that there were discussions about the industry / education gap from 1982 to 1996, when this topic virtually disappeared for the next 18 years while topics about technology in education soared by comparison. Conference paper topics shifted to technology as tool, rather than a job skill to be taught.

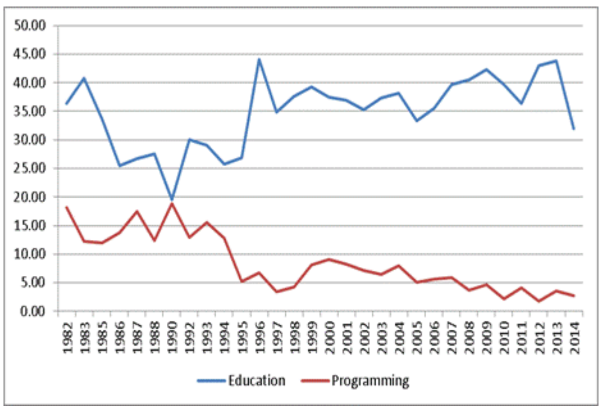


Figure 3: Programming/Education (As a percent of total KW's by year)

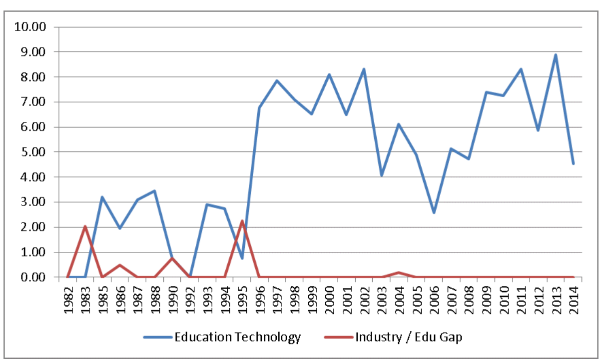


Figure 4: Industry / Education gap and Education Technology (As a percent of total KW's by year)

IT Skill/Programming Trends

A prevalent question in both the business and academic worlds of technology is "What programming languages are in demand?". Business needs graduates with these skills to fill new positions in an ever expanding field, and educators want to teach the correct "in demand" skills to their students.

2015	**2014**	**2013**	**2012**	**2011**
Java	Java	SQL	Java	Java
Java Script	C	Java	XML	C
C#	C++	HTML	Java Script	C++
PHP	C#	Java Script	HTML	C#
C++	Python	C++	C#	Java Script
Python	Java Script	C#	C++	Perl
C	PHP	XML	AJAX	PHP
SQL	Ruby	C	Perl	Visual Basic
Ruby	SQL	Perl	PHP	Python
Objec-tive C	MatLab	Python	Python	Ruby
2010	**2009**	**2008**	**2007**	**2006**
Java	Java	PHP	Java	Java
C#	C	C#	C	C/C++
C/C++	C++	AJAX	C++	C#
Java Script	Visual Basic	Java Script	PHP	Perl
Visual Basic	PHP	Perl	Visual Basic	Java Script
PHP	C#	C#	Perl	Visual Basic
Objec-tive C	Python	Ruby	Python	PHP
Perl	Perl	Java	C#	AJAX
Python	Delphi	Python	Java Script	Python
Ruby	Java Script	Visual Basic	Ruby	Ruby
Sources: "10 programming languages" (2015), Cass (2014), Hein (2013), Finley (2012), Sheenan (2011), Taft (2010), Connolly (2009), "10 computer Programming languages" (2008), "Most Popular Languages" (2007)				

Table 4: Top 10 programming language Lists by Year 2006-2015 (See sources.)

Table 4 shows the top 10 "in demand" programming languages from various sources over the last ten years. Java and variants of C (C, C++, C#) have been in demand all ten years, with PHP, Python and JavaScript in all but one or two lists.

With this consistency of demand across a decade and multiple sources, it should be easy for educators to see what they should be discussing and teaching. Below, Table 5 shows many of programming languages from Table 4 as they appeared as topics at ISECON.

Programming Language	First Year	Last Year	Total Keywords
C++	1994	2011	44
Java	1999	2013	27
Visual Basic	1995	2011	16
XML	2000	2008	9
HTML	1995	2001	7
C	1984	2006	6
AJAX	2006	2009	4
ASP.Net	2004	2007	4
C#	2004	2009	4
Python	2005	2011	3
JavaScript	2000	2004	3
Perl	1998	2009	3
PHP	2010	2010	1
Ruby	None	None	0

Table 5: ISECON Appearances of Top in Demand Programming / Scripting Languages

While C++ and Java are the top two by appearances, the third entry, Visual Basic is over represented compared to others in the list. C and C# have only 6 and 4 appearances at the conference, respectively, despite their consistent demand (Table 4), and C was not mentioned at ISECON after 2006, but is in the number 2 spot for demand for 4 years after 2006. PHP and Python seem quite neglected at ISECON, with only 4 keywords combined, but make the top 10 lists virtually every year. Perl is underrepresented with only 3 papers mentioning it, but was on the in demand lists for 7 out of 10 years. Ruby was left out in the cold at ISECON with no keywords, and yet was on the top 10 most demanded skills for 6 of the last 10 years. This comparison shows a very clear gap between what programming skills educators are talking about and what skills are in high demand with businesses.

Web-based programming and scripting skills are also in high demand. Some of these appeared in the top ten programming lists above. Table 6 shows 10 web specific skills that appeared in ISECON papers. Note the very low keyword counts. Web programming and website development are high demand skills and yet have very low representation at ISECON possibly because of the nature of ISECON over time.

Web development Technology	First Year	Last Year	Total Keywords
AJAX	2006	2008	4
Apache	2010	2010	1
ASP or ASP.NET	2001	2007	6
CGI	1996	1996	1
XML	2000	2008	9
HTML	1995	2001	7
JavaScript	2000	2004	3
PHP	2010	2010	1
XHTML	2002	2002	1
XSL or XSLT	2002	2006	4

Table 6: Web-based Programming Skill List

Figure 5 illustrates the overall trend of the Programming topic, both traditional and Web-based, over the entire 30 year period. Figure 5 isolates two clusters of keywords, one capturing programming in general and the other on web topics. Web topics began to appear in 1994 as one would expect. Although there are year to year variations, the coverage of web topics remains relatively consistent in the 2% to 8% range. The most surprising result is the significant drop in programming topics from a high of 19% in 1990, to around 3% by 2014.

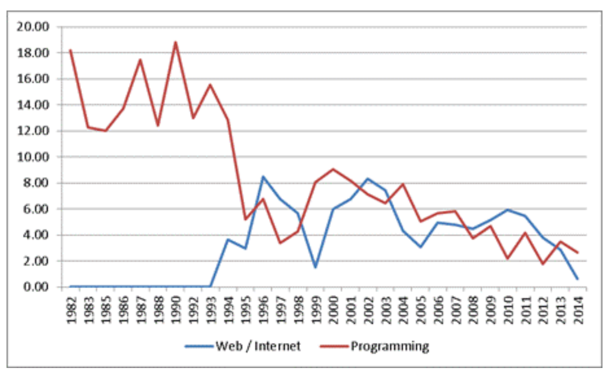


Figure 5: Programming/Web (As a percent of total KW's by year)

In 1993 or 1994 a significant decrease in traditional programming topics occurred, reducing the proportion from an average of 14% for a decade, to a mere 3% by 1997. Perhaps the development of the World Wide Web, and HTML programming displaced traditional programming, but even adding the two categories together, there is still a significant reduction.

Tables 7 and 8 contain three lists each from three different sources for the years 2009 and 2015, respectively, of the "Top 10" in demand I.T. skills for each year. This snapshot of skills six years apart was chosen to show what a student graduating from High School in 2009 might see as "in demand" skills and what educators might see as a guide to setting their curriculum for the next few years. Once the student finishes college in the typical five years, he or she would be looking for a job in the market portrayed by the 2015 table.

Skill List 1 (Marson, 2009)	Skill List 2 (Johnson, 2009)	Skill List 3 (Hoffman, 2009)
Business Process Modeling	Adobe Flash	Business Intelligence
Data mining	Article Writing	Data Center
Database	Content Writing (Web)	Help Desk Technical Support
IT architecture	CSS	Networking
IT optimization	Graphic Design	Programming Application Development
IT security	HTML	Project Management
Messaging Communi-cations	MySQL	Security
Networking	Photo-Shop	Telecommuni-cations
Project management	PHP	Web 2.0
Web development	Word- Press	(Only nine in list)

Table 7: Top I.T. skills for 2009

Note how different the skills are between the two years (2009 and 2015) as well as within one year. There is little agreement on the top I.T. skills, but one that stands out is data / database work, although it would be difficult to decide on specific skills to learn / teach based on these lists for both students and educators. This may explain some of the Industry / Education gap. Some of these "skills" are related to more applied skills like networking and others to more IS development skills such as programming. Some skills are very narrow single application or non-education skills like "Security Clearance (Federal - Active)" or "Share Point" that one would really just attain on the job. The lists contain a combination of very general and highly specific skills that would be near impossible to define or include in a college curriculum that is designed to be stable for a number of years.

Skill List 1 (Pratt, 2015)	Skill List 2 (Wadlow, 2015)	Skill List 3 (10 Fastest, 2015)
Big data	Cloud Security	.NET
Business intelligence analytics	Data Management	C#
Database administration	Data Science	Database administrator
Help desk/technical support	Enterprise Architecture	Java
Mobile applications and device management	Hacking	Oracle (applications)
Networking	Mobile Application Development	SAP
Programming application development	Network Penetration	Security
Project management	NoSQL	Security Clearance (Federal - Active)
Security compliance governance	Secure Coding	Share Point
Web development	Virtualization	Software developer

Table 8: Top I.T. skills for 2015

Figure 6 shows the trends for IS Development such as programming and database design and non-development IS skill groups as presented at

ISECON. If discussions about CS are overtaking discussions about CIS/MIS (as shown above in Figure 2) one might expect IS Development skills (ex. programming, database design) to be overtaking Non-development skills, and if CIS/MIS topics were more prevalent, Non-development IS skill discussions should be increasing. However, Figure 6 shows that Development skill topics are in decline and Non-development IS skill topics remain low and steady, indicating an overall trend away from covering technology skills of all types at ISECON.

Another example of this divergence can be seen in the keywords that cover computer/server operating systems. Table 9 shows occurrences of common operating systems in decreasing order of frequency in the proceedings. "Open source" has the most mentions, although LINUX is not mentioned once since its inception in the early 1990's. The second operating systems is the most interesting, but perhaps not surprising - Apple.

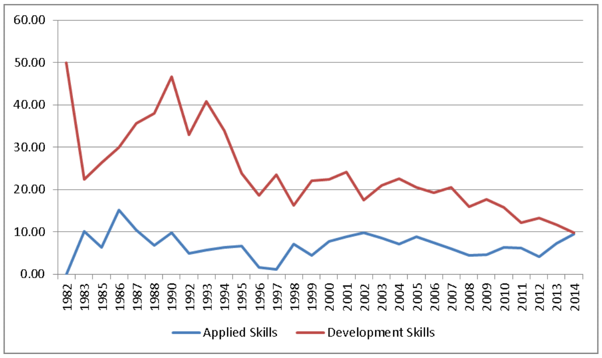


Figure 6: Applied vs IS Development skills

Apple markets strongly to the education community - students, institutions, and faculty. The problem here is that Apple has very low penetration into the business market, so "Apple" skills are niche and not in high demand (Table 10). This represents an education-centric view which puts emphasis on technology that is high in educator/student use, but low in "hirable" paid skills. This represents a "we are the world" education-centric view which puts emphasis on technology that is high in consumer use, but low in "hirable" paid skills. Windows seems to be underrepresented since it is the most common business desktop and server platform. Unix and DOS seem relatively in line, but having NO direct mention of LINUX seems very out of touch, since it is the most common web server installation in use. Android is arguably the most common mobile device OS, and yet it only has one entry.

Overall, the number of operating system entries seems quite low. Operating systems are the basis of all computer systems, and only 33 keywords out of 8390 directly reference an operating system, 0.4 percent of all keywords.

Operating System	First Year	Last Year	Total Keywords
Open Source	2005	2012	12
Apple/Mac	1990	2012	8
Windows	1994	2006	5
UNIX	1992	2007	4
DOS	1990	1994	3
Android	2012	2012	1
Linux	None	None	0

Table 9: Operating Systems

Desktop O/S	Percent market Share
Windows	87.8
Linux	1.7
Apple	7.5
Other	2.5
Web Server O/S	
Windows	32.9
Linux	67.1
Other	< 0.1
Mobile O/S	
Android	82.8
iOS (Apple)	13.9
Windows	2.6
Others	0.7

Table 10: Market Share
(Sources: "Desktop Operating System Market Share" (2015), "Global Market Share Held …" (2015), "Smartphone OS Market Share, 2015 Q2" (2015), "Usage of operating systems for websites" (2015))

Social I.T. Trends

Despite the dependence on the World Wide Web for social and business activities, not much attention seems to be paid at this conference to the tools needed to develop websites. Even if CIS students are not going to be web developers, one could argue that they need to at least understand what goes on behind the scenes from a technology standpoint, as well as the relationship to e-commerce and the underlying data that is collected by various online social and commercial websites. Table 11 shows several social media related topics.

Social Media Topics	First Year	Last Year	Total Keywords
Social Networking	2008	2013	11
Social Media	2011	2014	6
Social Networking and SNS	2005	2011	8
Blogs / Blogging	2004	2011	8
Facebook	2011	2014	3
Twitter	2009	2011	3

Table 11: Social Media Topics

As could be expected, social media and its uses began to be discussed in 2004 and their popularity has continued. The topics have included the general as well as the specific applications of social media with technology (Facebook, Twitter).

Social Career topics (Figure 7) appear nearly every year, but show a distinct increase in the 1990's and again in the 2000's after a drop off in 2002, with a peak in 2006 followed by another decline. A variety of different topics are discussed with the largest number of papers discussing "outsourcing" from 2005 through 2008.

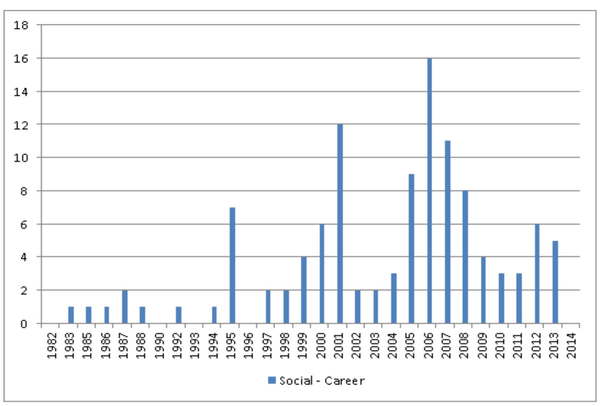


Figure 7: Social Career Keywords by Year (As total keywords)

5. CONCLUSIONS

CIS/MIS is a discipline that has seen a significant evolution over the past fifty years or so. This paper analyzed topics presented at ISECON from 1982 through 2014 to look for trends in the topic coverage as a surrogate for what is important in the IS field of education. The decline in technical topics, especially programming and web development, seem the most significant. Based on the findings here, these technical topics appear to have been totally replaced by teaching methodology than rather than technical skill content. Is this because these topics are "old" technology? Are they not interesting any ore to researchers who present at this conference?

These findings should encourage the discussion of what is important in teaching Information Systems – both topics and techniques – as well as what is important in the industries who hire new graduates.

6. REFERENCES

"10 Computer Programming Languages You Might Want to Know" (2008). QuinStreet. Retrieved June 24, 2015 from Source 2008: http://www.computer-schools.org/software/top-10-computer-programming-languages/.

"10 Fastest-Trending IT Skills: Spring 2015", (2015). IT Business Edge. Retrieved June 28, 2015 from http://www.itbusinessedge.com/slideshows/the-10-fastest-trending-skills-spring-2015.html.

"10 programming languages you need to know in 2015", (2015). Times of India Retrieved June 24, 2015 from Source 2015: http://timesofindia.indiatimes.com/tech/slideshow/10-programming-languages-you-need-to-know-in-2015/itslideshowviewall/46286489.cms.

Cass, Stephen (2014). Top 10 Programming Languages. IEEE Spectrum. Retrieved June 24, 2015 from Source 2014: http://spectrum.ieee.org/computing/software/top-10-programming-languages.

Connolly, Shaun (2009). Programming Language Popularity. spring.io Retrieved June 24, 2015 from Source 2009: https://spring.io/blog/2009/02/23/programming-language-popularity.

"Desktop Operating System Market Share" (2015). Net Market Share. Retrieved Sept. 2nd 2015, from https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0

Finley, Klint (2012). 5 Ways to Tell Which Programming Languages are Most Popular. readwrite. Retrieved June 24, 2015 from Source 2012: http://readwrite.com/2012/06/05/5-ways-to-tell-which-programming-lanugages-are-most-popular.

Foundation for Information Technology Education (2015). ISECON Proceedings. Retrieved March 31st, 2015 from http://proc.isecon.org/.

"Global Market Share Held by Operating Systems Desktop PCs from January 2012 to June 2015" (2015). Statista. Retrieved Sept. 2nd 2015, from http://www.statista.com/statistics/218089/global-market-share-of-windows-7/

Hein, Rich (2013). Top 10 Programming Skills That Will Get You Hired. CIO. Retrieved June 24, 2015 from Source 2013: http://www.cio.com/article/2368943/careers-staffing/97819-Top-10-Programming-Skills-That-Will-Get-You-Hired.html#slide1.

Hoffman, Thomas (2009). The 9 Hottest IT Skills for 2009. Computerworld. Retrieved June 27, 2015 from http://www.cio.com/article/2431399/careers-staffing/the-9-hottest-it-skills-for-2009.html.

Johnson, Dave (2009). The Top IT Skills in 2009. MoneyWatch. Retrieved June 27, 2015 from http://www.cbsnews.com/news/the-top-it-skills-in-2009/.

Davis, G., Gorgone, J.T., Couger, J.D., Feinstein, D.L. and Longenecker, H. E. Jr. (1997). IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, ACM, New York, NY and AITP (formerly DPMA), Park Ridge, IL, .

Marsan, Carolyn D (2009), Top 10 Tech Skills in a Downturn. Network World. Retrieved June 27, 2015 from http://www.pcworld.com/article/162324/top_10_tech_skills_in_a_downturn.html.

Most Popular Programming Languages (2007). DevTopics. Retrieved June 24, 2015 from Source 2007: http://www.devtopics.com/most-popular-programming-languages/.

Pratt, Mary K. (2014), 10 hottest IT skills for 2015. Computerworld Retrieved June 28, 2015 from http://www.computerworld.com/article/2844020/10-hottest-it-skills-for-2015.html.

Sheehan, Tom (2011), The Top 10 Programming Languages for 2011. myITview Retrieved June 24, 2015 from Source 2011: http://myitview.com/it-management/the-top-10-programming-languages-for-2011.

"Smartphone OS Market Share, 2015 Q2" (2015). IDC. Retrieved Sept. 2nd 2015, from http://www.idc.com/prodserv/smartphone-os-market-share.jsp

Taft, Darryl K (2010), Application Development: Top 10 Programming Languages to Keep You Employed. eWeek Retrieved June 24, 2015 from Source 2010: http://www.eweek.com/c/a/Application-Development/Top-10-Programming-Languages-to-Keep-You-Employed-719257.

Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker, J., Sipior, J., de Vreede, G. (2010). Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. http://www.acm.org/education/curricula/ IS 2010 ACM final.pdf

"Usage of operating systems for websites" (2015). W3Techs. Retrieved Sept. 2nd 2015, from http://w3techs.com/technologies/overview/operating_system/all

Wadlow, Thomas (2015), Top 10 IT Skills in Demand for 2015. Business Review Europe Retrieved June 28, 2015 from http://www.businessrevieweurope.eu/technology/380/Top-10-IT-Skills-in-Demand-for-2015.

Appendix (Additional Tables and Figures)

Categories	Keywords		Categories	Keywords
Bus-Company	13		Prog-Concept	112
Bus-Discipline	10		Prog-Game	16
Bus-Edu relationship	28		Prog-General	94
Bus-Education Gap	7		Prog-IDE	6
Bus-Experience	41		Prog-Language	34
Bus-General	85		Prog-Object Oriented	67
Bus-Management	48		Prog-Scripting	14
Bus-Model	10		Skills	73
Bus-Organization	7		Skills-Communication	23
Bus-Process	61		Skills-Computer	54
Bus-Project-Management	64		Skills-Group	98
Bus-Skill	6		Skills-IT	45
Bus-Technology	51		Skills-Math	38
Bus-Tool	5		Skills-Writing	35
Bus-Training	21		Social	120
Bus-Type	60		Social Media	49
Data-Analysis	59		Social-Career	114
Data-Dbase	98		Social-Ethics	103
Data-Dbase-Development	38		Social-Game	3
Data-Dbase-Language	29		Social-Human Factor	197
Data-Dbase-Online	3		Social-International	74
Data-General	88		Social-Law	52
Data-Management	21		Social-Privacy	16
Data-Modeling	65		Software	36
Edu-Accreditation/Org	153		Software-Application	50
Edu-Certification	41		Software-Computer Aided	47
Edu-Cheating	20		Software-Development	207
Edu-Course	200		Software-Game	8
Edu-Curriculum	734		Software-Graphics	14
Edu-Degree	74		Software-Model	50
Edu-Distance Teaching	276		Software-OS	12
Edu-Enrollment	76		System Administration	5
Edu-Faculty	31		System-Analysis	121
Edu-General	227		System-Architecture	31
Edu-Grading	23		System-Decision Support	52
Edu-Graduate School	23		System-Development	69
Edu-Graduation	9		System-General	87
Edu-High School	8		System-Security	137
Edu-Learning	178		System-Specific	99

Edu-Materials	30		Technology-Communication	30
Edu-Performance	62		Technology-Development	23
Edu-Project	56		Technology-General	92
Edu-Research	42		Technology-Hardware	33
Edu-Standards	19		Technology-Malicious	17
Edu-Student	121		Technology-Management	5
Edu-Teaching	374		Technology-Media	61
Edu-Technology	156		Technology-Mobile	39
Edu-Testing	95		Technology-Storage	10
Edu-Undergraduate	20		Technology-Support	2
General	561		Technology-Telecom	15
Government	18		Technology-Use	102
IT-Discipline	213		Technology-Virtual	24
Networking	104		Web-Application	22
Networking O/S	11		Web-Commerce	63
Networking-Server	18		Web-Development	96
Prog-1GL	2		Web-General	89
Prog-2GL	1		Web-Information	7
Prog-3GL	100		Web-Infrastructure	25
Prog-4GL	71		Web-Media	7
Prog-AI	18		Web-Service	44
Prog-Application Interface	39			

Table 12
Full Category table with total Keyword counts per category

Full Sized Figures for Greater Detail by Year

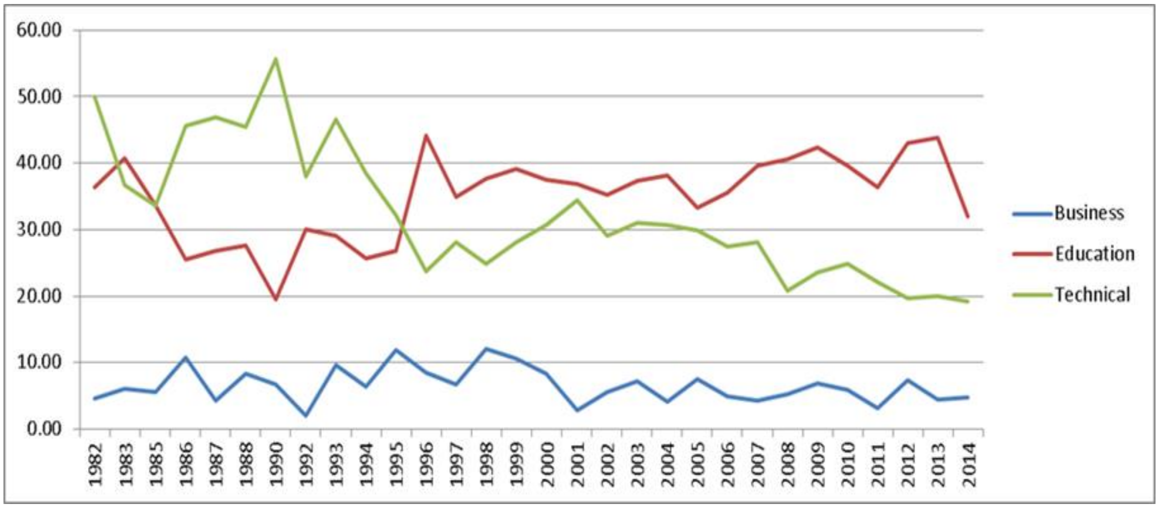


Figure 1: Technical/Education/Business (as a percent of total KW's by year)

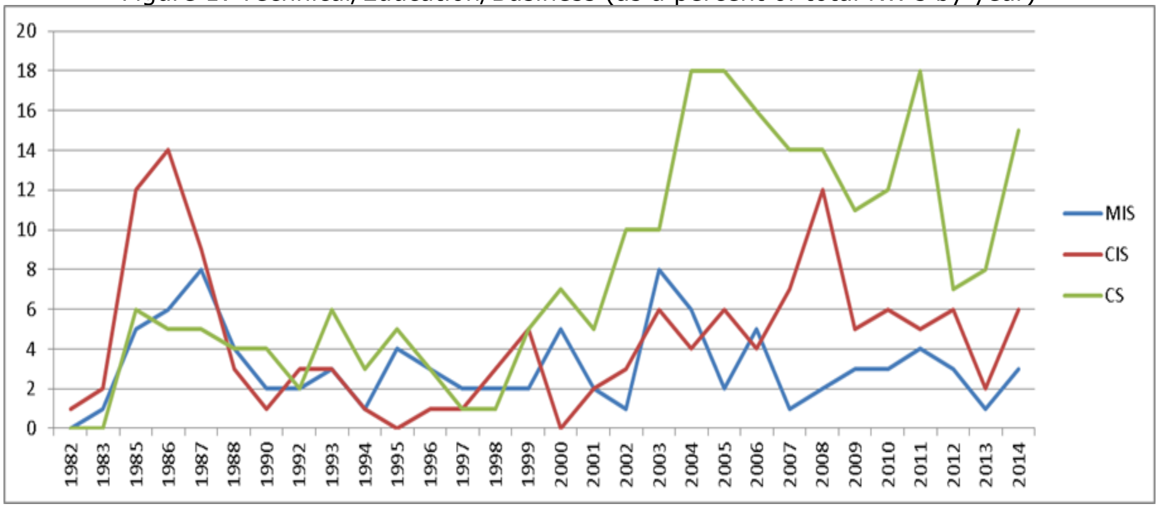


Figure 2: MIS/CIS/CS keywords (As actual number of keywords)

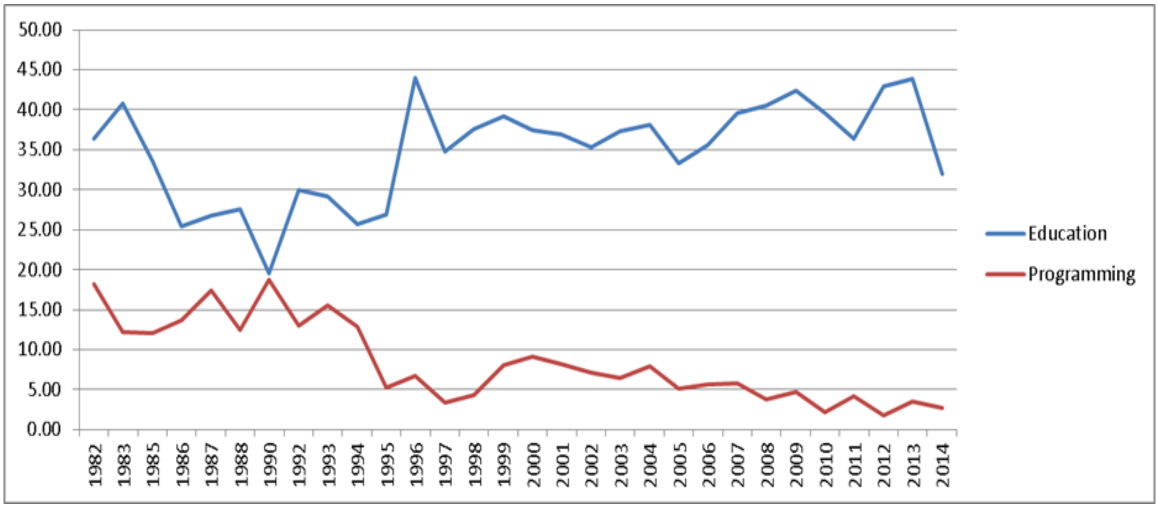


Figure 3: Programming/Education (As a percent of total KW's by year)

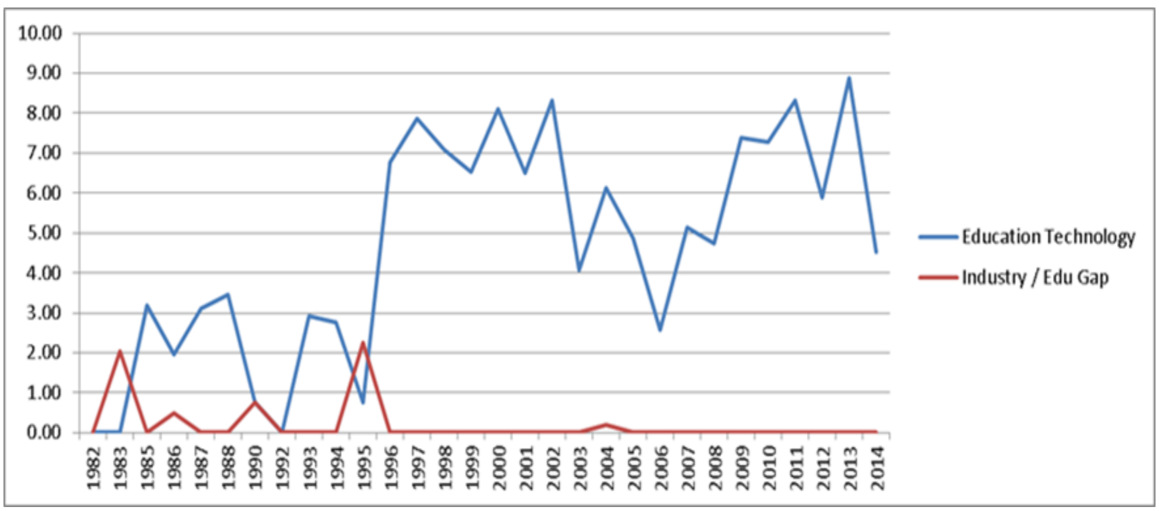


Figure 4: Industry / Education gap and Education Technology (As a percent of total KW's by year)

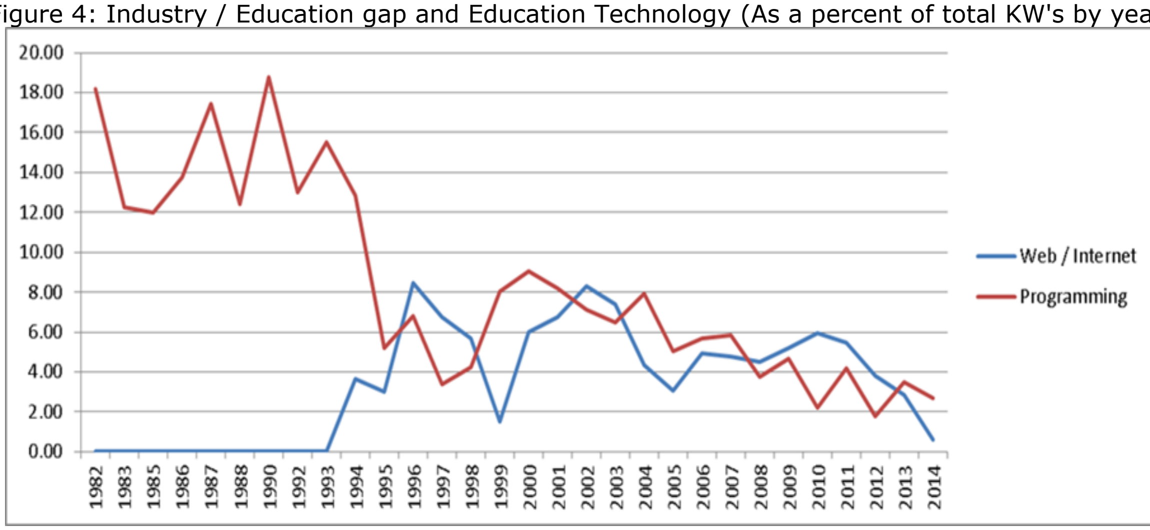


Figure 5: Programming/Web (As a percent of total KW's by Year)

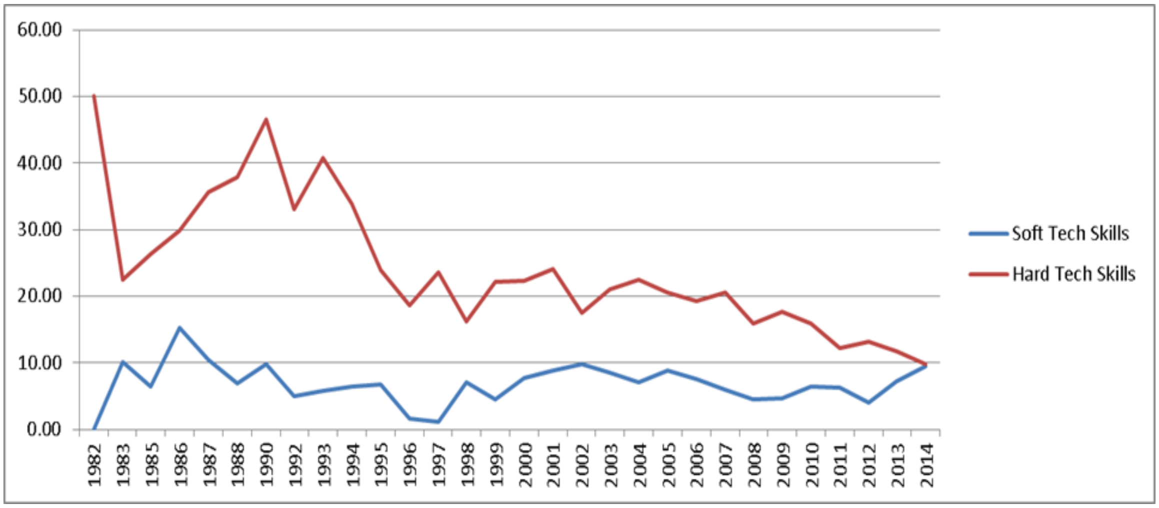


Figure 6: Hard vs. Soft Technical skills (As a percent of total KW's by Year)

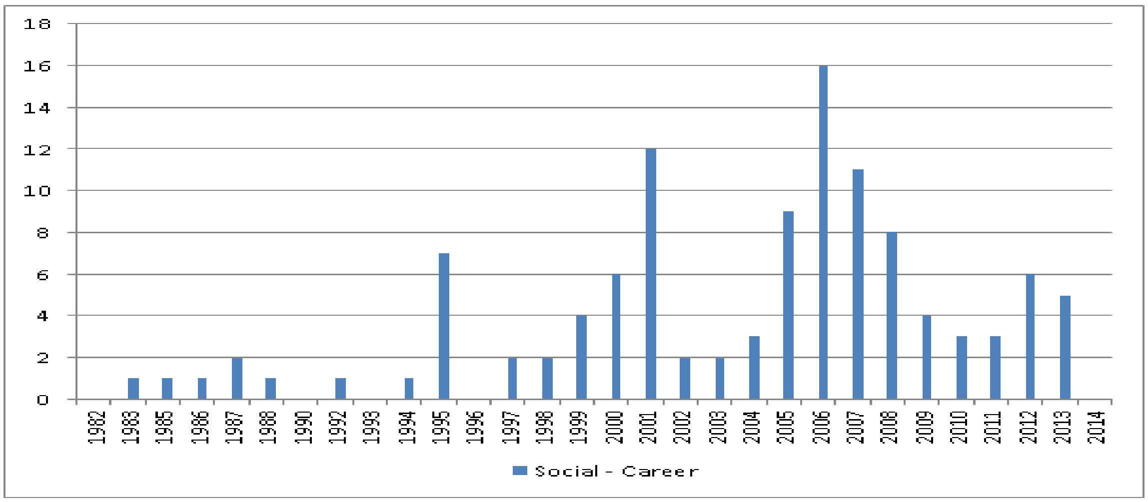


Figure 7: Social Career Keywords by Year (As total keywords)

Themed Learning with Music and Technology

Debra L. Smarkusky
dls102@psu.edu
Information Sciences and Technology

Sharon A. Toman
sat11@psu.edu
Music Department

Penn State University
Dunmore, PA 18512, USA

Abstract

Interdisciplinary teaching and interpretation of technology for various applications provides a challenging and engaging environment for students to enhance their creativity, critical thinking and problem-solving skills. This paper presents results of a joint effort between faculty in the Department of Information Sciences and Technology and the Department of Music to develop projects at various levels of the curriculum that utilizes a themed learning approach to enhance learned concepts. Students utilize technology as a means to convey their creativity, artistic design, and appreciation of the Arts. Since the creation of digital musical scores, animations and software applications share a similar development process, active-learning exercises using a variety of technologies, provide students with an opportunity to enhance learned concepts in both disciplines. While experiencing the project development process, students are learning object-oriented terminology, animation frameworks, computer programming, distributed computing concepts, and principles of music theory.

Keywords: Interdisciplinary Projects, Java, Music, LEGO MINDSTORMS NXT, Sibelius, Flash, Alice

1. MOTIVATION AND RELATED WORK

Because many of the most pressing problems of the day are best solved using interdisciplinary approaches, it is important that students be well educated in their own disciplines and at the same time be prepared to engage in interdisciplinary projects (Amoussou, Boylan, Peckham, 2010). The coalescence of arts and technology allows innovative research opportunities while upholding the essential principles of general arts education (Izmirli & Baird, 2002). Within academia, faculty are seeking to include interdisciplinary projects with challenging computational components to enhance collaboration between computer science and other disciplines such as engineering, physics, chemistry, arts, and the social sciences. Building bridges between academic areas encourages students to explore, build hypotheses, experiment, development critical thinking for problem solving (Barr, Liew, & Salter, 2010).

Music compositions and software applications share an analogous project development process. Because software processes are complex and there is no ideal process, many organizations tailor the fundamental activities to match their needs while enhancing the quality of software and performance of developers (Sommerville, 2016). While musicians use music notation to design and interpret musical concepts, software developers use designs to convey architecture and algorithmic processes for programs. Similar to the object-oriented paradigm in software development, design patterns were compared to the various conventions of music notation and how to incorporate them into the design of music notation software (Brandorff, Lindholm & Christensen, 2005). An interdisciplinary music and animation course with various technologies for object-oriented programming, music notation and animation presented a project development process (requirements, design, implementation and test/debug phases) that exists between disciplines (Smarkusky & Toman, 2009).

Computing technology can be utilized to enhance our creativity within the learning process. Students should have opportunities to engage in interdisciplinary projects where they can build confidence and enhance their learning. Barker completed an interview study of both faculty research mentors and undergraduates that analyzed the organizational, social and intellectual conditions under which undergraduate research was being conducted (Barker, 2009). Barker observed that undergraduate research projects in science, technology, engineering, and mathematics (STEM) have benefits such as "improved retention in both the major and discipline-related careers; ability to work independently and to communicate well with a team; increased confidence in academic knowledge and technical skill; broader awareness of the discipline; and awareness of career opportunities and support for making career choices" (Barker, 2009).

Replacing the traditional introductory computer science course in the curriculum with an interdisciplinary and "connected" pair of courses in creative arts, humanities, history, math and computer science, natural sciences, and social sciences has resulted in an increase in female enrollments, retention in computing, and new energy for interdisciplinary research opportunities. Connecting with Art courses in the curriculum includes design (creativity) and development (computational thinking) (LeBlanc, Armstrong, & Gousie, 2010). An interdisciplinary project that requires computer science and music majors to develop a notation for playing a non-musical instrument resulted in better appreciation of the importance of musical notation and software syntax in both disciplines (Heines, Greher & Kuhn, 2009).

Students majoring in computer science, mechanical engineering, interactive multimedia and music designed and developed "Conducting Robots", where a robotic and graphical conducting system direct an orchestra and provided an opportunity for students to focus on critical thinking, creative problem solving, and computational thinking skills (Salgian, Ault, Nakra, Wang & Stone, 2011). Glasser asserted that: being active while learning is better than being inactive. Most people learn only 10% of what they read, 20% of what they hear, 30% of what they see, 50% of what they see and hear, 70% of what they talk over with others, 80% of what they use and do in real life and 95% when they teach someone else (Glasser, 1998).

In the remaining sections of this paper, we specify the technologies that were selected for the completion of our interdisciplinary projects which include Alice, Flash, Sibelius, LEGO® MINDSTORMS® NXT, and leJOS. We then provide a summary of the projects that utilize these technologies and discuss the scaffolding of knowledge and support for enhancing learned concepts. We provide student comments about these projects and conclude with student feedback and summary.

2. MUSIC/ANIMATION TECHNOLOGIES

Alice

Alice (www.alice.org) is a freely available programming environment for the teaching of object-oriented concepts in introductory programming languages that was created and distributed by Carnegie Mellon University (Cooper, Dann, & Pausch, 2000). The Alice interface, shown in Figure 1, provides informative

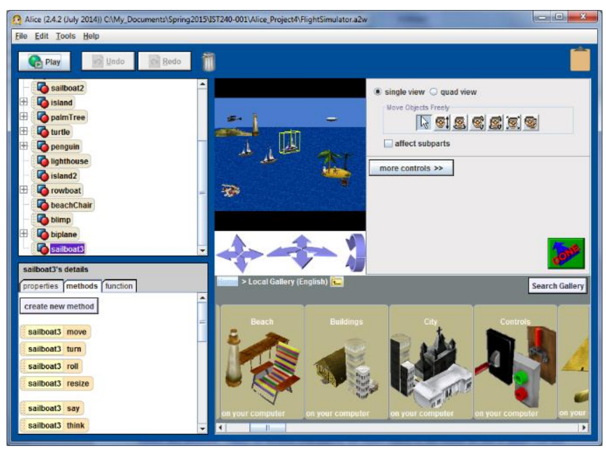


Figure 1. The Alice Interface

tutorials, a library of graphical components with embedded properties, methods and properties, camera controls, action events, and the ability to incorporate sound files all within an easy-to-learn drag-and-drop interface for the development of animations.

Flash
Adobe Flash Professional (www.adobe.com) is a development environment for creating animations and multimedia content. This platform provides students with the ability to develop interactive media that focuses on design aspects such as scenes, frames, and timelines and incorporates vector graphics, audio, images and scripting during development of media-enhanced projects. An example of the Flash Interface is shown in Figure 2. This tool provides

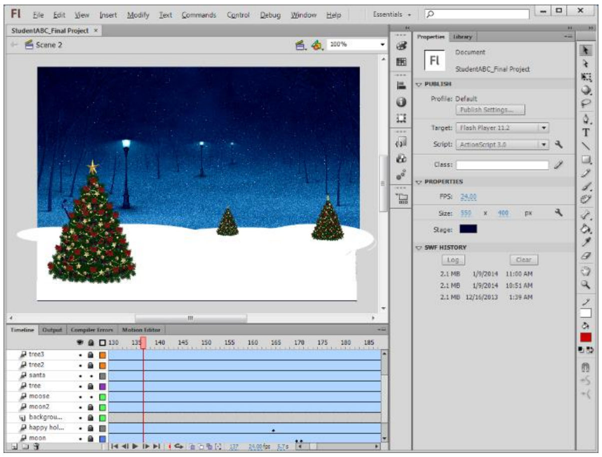


Figure 2. Flash Developer Interface

students with the ability to enhance their design experience via graphic layers and frameworks to better prepare them for the design and development of Java FX projects.

Sibelius
Sibelius (www.avid.com/US/products/sibelius**)** is a music notation application that is used by composers, arrangers, publishers, educators, and students to compose and write music. Sibelius, shown in Figure 3, provides users with a click-and-drag format for the insertion of musical notes and allows for instant playback of the musical score. This software application contains editing

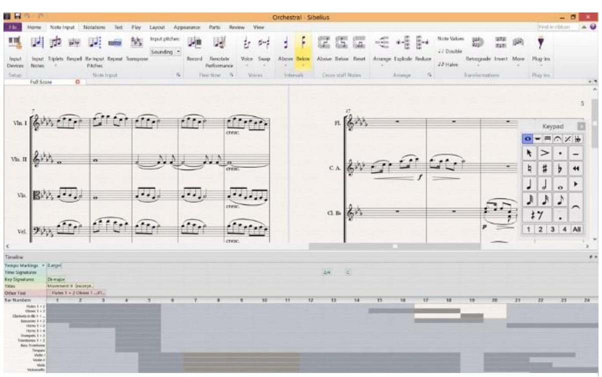


Figure 3. Sibelius Music Notation Application

tools for the management of notes, insertion of text and graphics, selection of musical instruments, and utilization of an assortment of built-in sound plug-ins.

LEGO MINDSTORMS NXT
The LEGO® MINDSTORMS® NXT platform, shown in Figure 4, provides the framework for students to experience software development using sensor input, multiple software threads of computation, and wireless communication. The

Figure 4. LEGO® MINDSTORMS® NXT

NXT brick includes four sensor ports ('1', '2', '3', and '4') and three input ports ('A', 'B', and 'C'). LEGO offers a wide variety of sensors to include a Sound Sensor, Light Sensor, NXT Color Sensor and Ultrasonic Sensor. Communication between the computer and the NXT brick can be realized with a USB cable connection or programmed to use the Bluetooth**®** (www.bluetooth.com) wireless communications. We elected to utilize this technology due to the abundance of available documentation, less expensive cost of the robot kits, and the extensible nature of the platform for various projects (LEGO MINDSTORMS Education NXT, 2015).

LeJOS NXJ
Originally created from the TinyVM project, which was an implementation of a Java Virtual Machine for LEGO® MINDSTORMS® RCX system, leJOS NXJ is a Java programming environment for the NXT (leJOS Team, 2009). This firmware is used to replace the NXT factory-loaded software, which includes a drag-n-drop environment for programming. This leJOS NXJ environment includes a Java Virtual Machine, a library of classes that implement the NXJ Application Programming Interface (API) for execution on the brick, a library of Java classes for computer programs that communicate with the brick via USB or Bluetooth, PC tools for debugging and

flashing the firmware, and the capability to compile, link and upload programs and other files to the NXT brick. The benefits of using leJOS NXJ was that it utilizes the Java programming language and can be developed using the NetBeans (www.netbeans.org) or Eclipse (www.eclipse.org) Integrated Development Environment (IDE) with associated plugins for both environments. Students enhance their problem-solving skills using object-oriented open source software to create challenging multimedia-based solutions.

3. INTERDISCIPLINARY PROJECTS

The content contained in our music and animation projects are presented in a scaffolding approach and utilize the seven principles for good practice in undergraduate education as presented by Chickering and Gamson as shown in Table 1 (Chickering & Gamson, 1987).

Encourages contacts between students and faculty.
Develops reciprocity and cooperation among students.
Uses active learning techniques.
Gives prompt feedback.
Emphasizes time on task.
Communicates high expectations.
Respects diverse talents and ways of learning.

Table 1. Principles for Undergraduate Education

Knowledge learned in earlier projects can be directly applied to concepts needed for future projects. Learning starts at the beginning of the curriculum when the basic animation and object-oriented concepts are introduced using Alice. This learning continues with advanced design and animation concepts in Flash and the composition of digital music scores in Sibelius. This knowledge provides the foundation for the design and development of more advanced interdisciplinary projects using Java via LEGO® MINDSTORMS® NXT platform and leJOS. All of these projects provide students with a better appreciation of the shared project development process that exists between programming, animation and music. In addition, these projects provide collaborative research and learning opportunities between disciplines, inspire ideas for advanced computing projects using multimedia, and can be utilized as a platform for interdisciplinary projects between technical and general education courses.

Introductory Java Course

Students in the Information Sciences and Technology (IST) program first experience interdisciplinary projects in the Introductory Programming Course, which can be completed in the first or second semester of the curriculum. Students are learning the object-oriented terminology for the first time and Alice provides a fun environment to familiarize students with these concepts. Students are introduced to the Alice World and the associated gallery that contains classifications for the types of objects that can be added to a scene. Once objects are instantiated (added to the scene), we discuss the properties (attributes) and behaviors (methods) for each object shown in the Object Tree. Students also learn how to add sound effects and background music to enhance their animations. Each project provides students with an opportunity to be creative while learning the importance of organizing concepts and behaviors. The creation of new methods allows students to reuse the code they had previously created in other areas of their animation, especially event-handling. Logical structures provide variable behaviors, while event-handling and camera controls provide a more interactive experience.

Alice provides the perfect learning environment for which requirements definition, design, implementation and test/debug activities could be introduced with instant visual feedback about their performance. The project description represents the requirements definition. Students identify various classifications from the gallery and determine how they would be used in the animation. For the implementation phase, students add instances of the objects to the world, position them as specified in their design, and utilize the drag-and-drop interface of Alice to create new methods, events and logical structures to realize the required behaviors and actions of their animation. Students select the play button to ensure correctness and completeness and make changes to the logic in methods or placement of the object within the scene. The Alice environment provides a good introduction to object-oriented terminology, software development phases and the utilization of sound files for special effects. Once students complete several projects using Alice, they are introduced to the Java programming language.

General Education Arts Course

The INART236 - Introduction to Music and Animation with Technology course is a General Education Arts course that students are encouraged to complete after the second semester. The fundamental guidelines for a

General Education course at our university states that the course "aids students in developing intellectual curiosity, strengthened ability to think, and a deeper sense of aesthetic appreciation" (Penn State University, 2012). In this course, Sibelius is utilized to aid in learning of music theory and the creation of a musical score, Flash allows students to realize more complex animation concepts, and a capstone project at the end of the course combines lessons learned in both disciplines (Smarkusky & Toman, 2009).

The first six weeks of the semester focuses on the basic fundamental concepts of music theory and composition using Sibelius. Learning the rudiments of music requires an understanding of each of the elements of music before they can sing, play or write music notation. The fundamental concepts of music theory is comprised of notation, scales, meter, rhythm, intervals; and basic chord structure. As students are learning each of these music fundamentals they are working with technology via the interface of Sibelius to solve problems through active-learning exercises. These exercises, which are created in Sibelius, challenge the student's ability and knowledge about each learned concept. The click-and-drag interface of Sibelius allows students to easily insert a musical note and its rhythmic value into a digital music score. In addition to learning concepts of music theory and composition, students gain knowledge about a wide variety of musical instruments that include strings, woodwinds, brass, percussion, and keyboard instruments. This knowledge provides students with the needed understanding to allow them to incorporate staves for each of the newly added musical instruments during composition. In addition, Sibelius provides additional plug-ins for adding drum patterns, guitars, rhythmic patterns, etc. The process of creating a musical composition from scratch is a time consuming process of adding notes for a simple melody, chords for harmony, articulation and expressive nuances as well as instruments. It is a step-by-step process that must be followed correctly for a successful digital musical score. Music has often been compared to learning a foreign language. Students must learn and have an understanding of the basic music theory terminology to be able to understand the more complex components required to complete a digital musical score.

Students then learn to develop more complex animations using Flash during the second six weeks of the course. The Flash framework provide students with an opportunity to design animations that include timelines, scenes, layers, and frames. Gridlines, rulers and guides provide guidance for the drawing of objects, creation of symbols or text components. Students learn how to create frame-by-frame animations, motion and shape tweens, and graphic symbol animations. The more advanced features include animated text, motion guide layers, mask layers, and the inclusion of sound. Interactive features include frame labels and buttons with ActionScript being utilized for more complex actions. Once again, students are gaining experience with the project development process (Smarkusky & Toman, 2009). To ensure that students are gaining experience using a specific animation component, project descriptions and expected results are provided for each exercise. During the design phase of the project, students need to identify various scenes, determine layers, and the animated effects that would occur at each frame of the timeline. Similar to creating graphical user interfaces for their software development courses, students need to focus on the design of their interface (scene) and layer components to achieve the desired result. Students design in a top-down approach and utilize a bottom-up approach for implementation. Each animation component is added to a layer, layers are added to scenes, and scenes are integrated to create the final animation. Students play the animation to check that all requirements were satisfied and make appropriate modifications when needed.

During the last three weeks of the course, students are assigned a capstone project that focuses on combining knowledge gained in both Sibelius and Flash. The goal of the project is to create an animated greeting with background music. For the music component of the project, the requirements definition is a simple and introductory piece of sheet music, which is provided to students for them to compose digitally using Sibelius. Students need to pay specific attention to detail to ensure that key signatures, voices, notes, accidentals, articulation, etc. match those in the requirements definition. Students are allowed to enhance the musical piece by adding additional staves for various instruments as well as utilize a software plug-in to create a percussion part to enhance the style of their composition. Once testing is completed, students are required to export the musical score and convert the file to an MP3 format for use in Flash.

For the animation component, students must satisfy the specified technical components of the animation while keeping within the theme of the assigned musical song. Technical requirements include the addition of a separate layer for incorporating the sound file that was created in Sibelius, creation of a new symbol, animated text block with individual letter animation, use of a mask layer, motion tweens, shape tweens, action frames and labels, and button implementation to include ActionScript. The open-ended nature of the animation allows students to gain experience in requirements definition and apply knowledge gained from active-learning and homework exercises to the design phase of the animation. Implementation and testing is an iterative process as students need to ensure the frame-by-frame execution of the project is correct to that point of completion. The assessment for this project is based on the correctness and completeness of their musical composition, their creativity in correlating the animation with the theme of the music, and the correct implementation of the technical components of the animation.

Advanced Java Course

During their junior year, IST students are required to complete a Distributed Computing course. One of the projects within this course is the LEGO Maze Project as demonstrated in Figure 5 (Smarkusky, Toman, Sutor & Hunt, 2013). Students gain experience with the leJOS interface

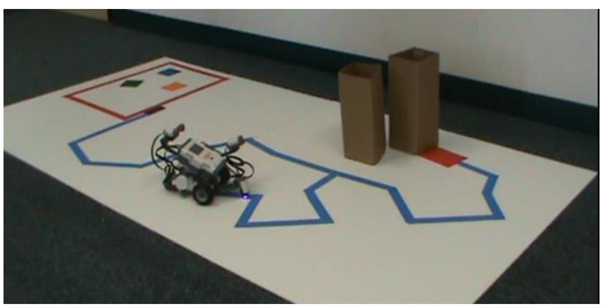


Figure 5. LEGO® Maze Project

using Java where they can compile, upload, and execute programs on the LEGO® MINDSTORMS® NXT. Unlike traditional software projects that obtain input from graphical user interfaces, files or databases, this project utilizes input from the sound, color and ultrasonic sensors of the robot. The values read from the sound sensors are impacted by environmental acoustics; robot motors react differently based on the flooring surface; color sensor values change with sunlight, shadows, natural or florescent lighting; and ultrasonic senor values are constantly changing due to range of vision and constant movement of the robot.

This project requires that students utilize user defined sound files or the onboard sound clips that are provided with the leJOS platform. In preparation for a future project, students must use the sound sensor to start the robot in motion (e.g. a hand clap that is at least 25 decibels). Once in motion, the robot must follow a specified path (Color1). Upon reaching the first wall, the robot should turn right 90 degrees and continue following the path. When it reaches the second wall with a different color (Color2), it will turn 180 degrees and continue following the Color1 path backwards, keeping on a straight path when possible. Upon reaching the sand box (Color3), the robot will turn left 90 degrees and begin a search pattern. The goal within the sandbox is to find three additional color swatches (Color4, Color5, Color6), and play a unique sound for each of the identified colors. The challenging nature of this project provides an opportunity for problem-solving, learning the LeJOS Application Programming Interface (API), and gaining additional experience with Java Threads.

Once familiar with the leJOS API, sensors, and the ability to maneuver the robot, IST students are assigned a project in which they must work with students from a General Education Music Theory course to create a robot dance where the movements of robot couples (two robots) are synchronized to the rhythm of the music (Smarkusky & Toman, 2013). With the choreography of the dance being the bridge between disciplines, the background music is created from scratch by the music students and the robot dance movements are implemented by the IST students.

Within the music theory course, students are introduced to more complex characteristics of melody, rhythm, harmony, and instrumentation. Similar to the General Education Arts course, students start with a more advanced piano musical score that is then created digitally using Sibelius. This project is further enhanced using various instrumental parts to include flute, trumpet, saxophone, guitar, bass guitar, and drum pattern plug-ins. Students are able to control tempo, dynamics, and other music nuances using the Sibelius's integrated *Playback* feature.

The IST students implement Java applications, utilizing the leJOS NXJ environment, to maneuver the robots based on the steps identified in the choreography which was also designed by the music students. This provides unique learning opportunity for the IST students as they realize the importance of threads and the completion of object-oriented methods for each routine with parameters for time, duration, direction, or rotation for various movements. The timeline that is included with the *Playback* feature of Sibelius provided assistance with synchronization of the robot dance movements to the music file. Assessment for this project is based on the correctness and completeness of the technical requirements of the project for each discipline along with the complexity of the choreography and how well the movements were synchronized to the music. Similar to the comparison presented by Do & Gross (Do & Gross, 2007), the programmer is not expected to be a music student, and the music student is not expected to be a programmer. Each member of the interdisciplinary team utilizes their strengths for the success of the project.

4. FEEDBACK AND SUMMARY

Overall, the results of incorporating a musical theme to allow students to learn technical concepts was a positive experience. At the freshman level, Alice provided a platform to excite students about music and animation, the project development process, and object-oriented terminology, as a preamble to the development of projects using Java. During the sophomore year, students were encouraged to enroll in the General Education Arts course in music and animation. Since the development of digital musical scores, animations, and software applications follow a similar project development process, the completion of this course provided students with opportunities to understand the importance of paying attention to details (within Sibelius and Flash), learning how the project development process can be applied to other disciplines, and providing students with an opportunity to learn more about designing interfaces using layers that will provide a foundation for the learning Java FX in future Java programming courses.

The General Arts Education course was offered in the Fall 2013 semester. This course had an enrollment of 20 with 14 students electing to complete an optional survey. As shown in Table 2 of the Appendix, and represented below with actual student comments, the students enjoyed using various technologies, had a better understanding of the project development process, increased their creative abilities, gained confidence with the application of learned concepts to other disciplines, and had an increased appreciation for the attention to detail and time involved with creating musical animations.

> *"Completing my music and animation project did change my perception of music, animation, and technology. It was pretty technical to make pieces of music or animation, yet putting it all together was a little difficult, yet challenging and fun."*

> *"When I look at music and animation, I see how much work goes into it and I enjoy it much more. I now see the bigger picture and see that you can get very creative with it if they are willing to learn."*

> *"Music is good for teaching logic and animation can teach patience. Also, they both can give students an edge with some jobs."*

> *"I think the arts are an underappreciated aspect of our culture and music and animation go hand in hand today. You don't have to go far to see some great achievements online using a combination of these. I would definitely say that getting an introduction into the complexity to gain appreciation for it is a good idea."*

Since students required more experience with Java to include concurrent threads of control and programming with limited resources, we elected to wait until the advanced Java course for the application of projects using LEGO® MINDSTORMS® NXT and the leJOS firmware. Assigning the LEGO Maze project as required project in the course, allowed students to gain experience and confidence with these new technologies prior to being paired with the music theory students for the completion of the LEGO Dance project. Using the leJOS programming to develop programs for the NXT can challenge one's software development, robotics, and programming skills. The LEGO Dance project incorporated a new perspective to the software development process in that students now needed to implement the requirements (choreography) that were designed by the music theory students, and be able to explain the limitations of the technology and the project (keeping two robots on a 6'x12' floor surface, implementing moves via two robot wheels and not actual human feet, etc.). The expectations of both the IST and music students were high, so they worked together as a team to complete and solve the challenges of this project. At the end of the project, we surveyed all students, both music and IST, to get their feedback. We have incorporated the LEGO Maze and LEGO Dance Project into the advanced Java course every spring semester since 2012. The combined feedback for the LEGO Dance Project over the past four years with 96 students participating is shown in Table 3 of the Appendix, and includes responses from both Music and IST students.

The results demonstrate that the LEGO Dance Project was a success. Students were able to express their creativity, enjoyed working with students from other disciplines, learned more about animation and recommended that this project be offered again in the future. The project teams consisted of three music students for every one IST student. We expected the response rate for learning more about music composition to be lower than other results. This is based on the fact that the music students were involved with both music and animation components of the project, as compared with the IST student who participated only in the animation component of the project and did not have any involvement with music composition. Some additional student comments about the LEGO Dance Project are shown below:

> *"It was cool to be able to make robots dance to the music we created, I never even would have thought of incorporating IT with music. I enjoyed the accomplished feeling at the end of the day to be able to put all of this together and make it work."*

> *"Being able to animate the robot was fun, but the limited physical abilities of the robot made it challenging and frustrating. Incorporating creativity in the project was accomplished when, as a group, we all discussed the limitations and had to creatively find a solution around these limitations that still looked appealing."*

> *"I think this project is a great way to test the knowledge of students and work in teams. It was a good experience meeting with IST students and choreograph the dance."*

> *"Working with the robots as opposed to the same IDE, GUI, and box is fantastic. It's a wonderful experience to have a "business" that gives "demands" that we need to deliver. Especially so when they don't 100% understand the constraints of the code and language so we need to act as intermediaries. This was an outside the box project and I truly enjoyed it."*

In summary, these interdisciplinary projects provided students in both Music and IST courses with an opportunity to become familiar with terminology and concepts from both disciplines. Students quickly learned that the creation of digital musical scores, animations, and the Java programs that controlled the choreography all required attention to specific detail and the following of a very similar development process. In identifying tools for application of concepts, both departments preferred software that would provide instant feedback to students during project development.

By using Alice, Flash and Sibelius, we found that students are trained to look for specific details that are required for the creation of a correct and complete deliverable in both music and animation. This attention to detail and shared project development process provides additional experiences to students for the development of Java applications and the utilization of leJOS NXJ firmware for the LEGO maze and LEGO Dance projects. The success of these projects and support from student feedback warrants the continued offering of these projects in the future. In addition, we have been asked to offer the General Education Arts course as part of a new honors program at the campus. Students who have participated in these projects requested that faculty offer undergraduate research opportunities in music and animation. We will continue utilizing the LEGO® MINDSTORMS® NXT and the leJOS firmware for current projects and plan to incorporate the new LEGO® MINDSTORMS® EV3 into additional undergraduate projects in the next academic year.

5. REFERENCES

Amoussou, G., Boylan, M., & Peckham, J. (2010). Interdisciplinary computing education for the challenges of the future. In *Proceedings of the 41st ACM technical symposium on Computer science education* (SIGCSE '10). ACM, New York, NY, USA, 556-557. DOI=http://doi.acm.org/10.1145/1734263.1734449

Barker, L. (2009). Student and Faculty Perceptions of Undergraduate Research Experiences in Computing. *Transactions on Computing Education* 9(1), Article 5. March 2009, 28 pages. DOI=10.1145/1513593.1513598 http://doi.acm.org/10.1145/1513593.1513598

Barr, V., Liew, C.W. & Salter, R. (2010). Building bridges to other departments: three strategies. In *Proc. of the 41st ACM technical symposium on Computer science education* (SIGCSE '10). ACM, New York, NY, USA, 64-65. DOI=10.1145/1734263.1734285 http://doi.acm.org/10.1145/1734263.1734285

Brandorff, S., Lindholm. M. & Christensen, H.B. (2005). "A Tutorial on Design Patterns for Music Notation Software", *Computer Music Journal*, 29(3), pp. 42–54.

Chickering, A. W. & Gamson, Z. F. (1987, March). Seven Principles for Good Practices in Undergraduate Education. *The AAHE Bulletin*. Retrieved on July 1, 2015 from http://www.aahea.org/aahea/articles/sevenprinciples1987.htm

Cooper, S., Dann, W. & Pausch, R. (2000). "Alice: A 3-D Tool for Introductory Programming Concepts", *Journal of Computing Sciences in Colleges*, 15(5):108-117.

Do, E.Y. & Gross, M.D. (2007). Environments for creativity: a lab for making things. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition* (C&C '07). ACM, New York, NY, USA, 27-36. DOI= http://doi.acm.org/10.1145/1254960.1254965

Glasser, W. (1998). *Choice Theory in the Classroom*, New York: Harper Collins, ISBN: 0-06-095287-3.

Heines, J., Greher, G. & Kuhn, S. (2009). "Music Performatics: Interdisciplinary Interaction", *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE),* pp. 478-482.

Izmirli, O. & Baird, B. (2002, May). "A Model for Integrating Arts and Technology in a Liberal Arts College", *Journal of Computing Sciences in Colleges*, 17(6), pp. 102-109.

LeBlanc, M.D., Armstrong, T. & Gousie. M.B. (2010). Connecting across campus. In *Proceedings of the 41st ACM technical symposium on Computer science education* (SIGCSE '10), ACM, New York, NY, USA, 52-56. DOI=10.1145/1734263.1734280 http://doi.acm.org/10.1145/1734263.1734280

LEGO MINDSTORMS Education NXT (2015). https://shop.education.lego.com/legoed/en-US/search/navSearchResults.jsp?categoryId=EDU_PRD_LINE_106&ProductLine=MINDSTORMS+Education+NXT, accessed July 2015

leJOS Team (2009). http://lejos.sourceforge.net/

Penn State University (2012). University Faculty Senate, Policies and Rules for Undergraduate Students, Baccalaureate Degree Curriculum, http://senate.psu.edu/policies-and-rules-for-undergraduate-students/appendix-a-1-general-education-baccalaureate-degree/, accessed July 2015.

Salgian, A., Ault, C., Nakra, T.M., Wang, Y., & Stone, M.K. (2011). Multidisciplinary computer science through conducting robots. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (SIGCSE '11). ACM, New York, NY, USA, 219-224. DOI=10.1145/1953163.1953229 http://doi.acm.org/10.1145/1953163.1953229

Smarkusky, D.L. & Toman, S.A. (2009). An interdisciplinary approach in applying fundamental concepts. In *Proceedings of the 10th ACM conference on SIG-information technology education* (SIGITE '09). ACM, New York, NY, USA, 224-228. DOI= http://doi.acm.org/10.1145/1631728.1631800

Smarkusky, D.L. & Toman, S.A. (2013). An Interdisciplinary Learning Experience: The Creation of a Robot Dance, *Information Systems Education Journal* 11, 5, pp 55-62.

Smarkusky, D.L., Toman, S.A., Sutor, Jr., P. & Hunt C. (2013). Performing robots: innovative interdisciplinary projects. In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education* (SIGITE '13). ACM, New York, NY, USA, 125-130. DOI=10.1145/2512276.2512285 http://doi.acm.org/10.1145/2512276.2512285

Sommerville, I. (2016). *Software Engineering*, Tenth Edition, Pearson, ISBN: 978-0-13-394303-0.

APPENDIX

Question (1=Strongly Disagree; 5=Strongly Agree)	Mean	Std Dev	Positive Responses (%)
I now understand the fundamental concepts of music theory and the technical components of animation for the creation of a musical animation	4.21	0.58	93
I enjoy using various technologies for the development of digital music scores and animations	4.21	0.58	93
I now understand the phases of the project development process that are necessary for the creation of digital musical scores and animations	4.29	0.61	93
During this course, I developed and applied skills in creative thinking through design and active learning exercises	4.29	0.47	100
I enjoyed creating a musical animation that encompassed all learned concepts, that provided an opportunity for the expression of creativity as well as interdisciplinary and integrated learning	4.07	0.73	79
I feel that the information learned in this course could enhance projects completed in other disciplines	4.14	0.53	93

Table 2. General Education Arts Course Feedback

Question (1=Strongly Disagree; 5=Strongly Agree)	Mean	Std Dev	Positive Responses (%)
This project was a creative learning experience	4.28	0.68	92
I enjoyed working with students in other disciplines	4.25	0.80	89
Both Music and IST student worked together as a team to create a successful and complete project	4.33	0.74	94
I learned more about music composition during this project	3.68	0.95	63
I learned more about animation during this project	4.19	0.81	84
I would recommend offering this project again to Music/IST students in the future	4.21	0.83	85

Table 3. LEGO Dance Project Feedback

Assessing Faculty Perceptions and Techniques to Combat Academic Dishonesty in Online Courses

Karen Paullet
paullet@rmu.edu
Robert Morris University

Adnan A. Chawdhry
chawdhry_a@calu.edu
California University of Pennsylvania
California, PA

David M. Douglas
douglas@rmu.edu

Jamie Pinchot
pinchot@rmu.edu

Robert Morris University
Moon Township, PA

Abstract

Online Education is growing as it provides an added convenience to students, especially ones who have life circumstances that prevent them from attending traditional classes. With this growing trend, faculty and universities are facing increased risks with validating student identities in online courses and combatting unethical student behavior. This exploratory study of 75 faculty members will examine faculty perceptions of academic dishonesty specific to validating student identities along with discussing techniques used to combat academic dishonesty. Additionally, it is important to understand how faculty perceptions correlate with demographic characteristics and their experiences in order for administrators and universities to effectively develop techniques to mitigate the risks associated with academic dishonesty in online courses.

Keywords: Distance learning, online learning, academic integrity, student verification

1. INTRODUCTION

Technology surrounds us during each of our waking and sleeping moments. Our nightly visits to the land of Hypnos and Morpheus no longer offer us a reprieve from the ubiquitous grip of electronic technology and the quest to stay perpetually connected to the world. It is no wonder that the instructors in the academic world are frequently challenged with the prospects of student cheating in the online assignment environment. Students and instructors alike are tethered to their electronic devices in our twenty-four-seven world. Most students are tied to their mobile devices for fear of missing a moment of connectivity to the outside world and to those whom they feel compelled to maintain an electronic relationship. Instructors in academia are no different and also wear the stigmata of an electronic chain.

One of the challenges of teaching and testing in the virtual world of academia is the preservation of the integrity of the online educational environment. As online instructors we often have to come to terms in which we are supposed to "believe" in a student's honesty and integrity. Most, if not all, post-secondary institutions have a code of conduct that the student aggresses to implicitly or in a signed agreement.

As universities have attempted to remain competitive by offering online courses, validating student's identities have become difficult. There is no doubt that distance learning has always seen this challenge and been questioned with validating the identity of distance education students. As a result, universities and colleges should focus on implementing verification methods to establish user identities. This study will assess the validation of student identities, as well as, authentication and academic integrity methods that are set in place for distance learners. The following research questions were explored:

RQ1: Is there a correlation between faculty authentication methods used in online courses with age and gender?

RQ2: Do university faculty adopt techniques to mitigate the risks of unethical behavior in online courses?

2. BACKGROUND

The validation of student's identities has been questioned by those critical of distance learning (Baile & Jorbert, 2008). Do educators really know who has completed the course requirements in online classes is a question of debate. The College of Opportunity and Affordability Act (H.R. 4137) , "requires institutions that offer distance education to have processes through which the institution establishes that the student who registers in a distance education course or program is the same student who participated in and completes the program and receives the academic credit" (Database, 2007). In order to maintain the integrity of online learning, validating the end users identity is a priority (Paullet, et.al, 2014).

According to Howell, Sorensen & Tippets (2010), many distance educators are concerned about the integrity of their courses but few are willing to spend time or resources to continually learn the newest cheating methods and techniques for prevention of them.

A study conducted by Grijalva, Kerkvliet, and Nowell (2002), examined the level of cheating in online courses. The researchers used class cheating and testing policies to examine student responses. The findings revealed that academic dishonesty in a single online class is no more prevalent than in traditional classrooms. Throughout education, students have always come up with excuses as to why their assignments are late. A familiar reason is "the dog ate my homework." In 2015, living in a world surrounded by technology, student excuses have changed. We now hear, "the dog ate my flashdrive," "the Internet connectivity dropped while I was taking an exam," or "my computer has a virus." Technology has provided students with new opportunities for academic dishonesty. Students can now search the Internet while taking an online exam, or use thoughts and references of published authors as their own during an exam or when completing assignments. The ease of cheating and plagiarism in online courses is less visible to their classmates and instructors.

3. DISHONESTY METHODS USED BY ONLINE STUDENTS

Students are now using new methods to cheat online. Krask (2007) provides an example where a student starts an exam so that they are able to view all of the questions. The questions are then printed so that the student has time to search for the answers. After printing the questions, the student then disconnects from the Internet which will cause the exam to lock, which in turn does not permit the student to enter the exam to finish. After searching for answers, the student emails the instructor, often times providing a screen shot of the connection loss from the Internet, and requests for the exam to be reset (Rowe, 2004).

Online students often also use a "waiting" approach. Many online instructors allow a few days to take an exam. Since the timeline is a bit flexible, some students may wait until others have already taken the exam, and then ask for the questions. While online testing software can often restrict the ability to print out questions, students can still take screenshots of the questions or photographs on their mobile phones.

Howell et al. (2010) note that one of the newest methods of cheating involves the use of "braindumps," which are actual online businesses that provide students with studying services and often guarantee passing scores. Some of these sites may even offer access to exam questions and homework solutions directly from the instructor versions of textbooks, as well as access to previously graded essays or assignments. Four of the most well-known braindump sites are Cramster, Koofers, Study Blue and Course Hero which are considered tutoring sites where students can review past exams, assignments and projects used in their current courses. However, braindumps can pop up in a variety of places, or even for very specific schools or courses. One professor from Indiana State University found her test questions for sale on eBay (Howell et al., 2010).

In addition to braindumps, students continue to utilize new technologies to enhance cheating activities. One of the current popular cheating methods is use of mobile phones. Students can keep crib notes on their phones, use text messaging to have a friend lookup answers for exam questions, or take photographs of an exam and transmit them to other students (Howell et al., 2010).

Students are also using online services such as WeTakeYourClass.com or BoostMyGrade.com in which they pay a fee for someone to take their online classes. Prices vary according to the complexity of the class and type of class. For example, a student will pay a higher fee for an advanced biology course than they would for an introduction to computers course. They can also pay for individual assignments or projects to be completed rather than the entire course.

4. PREVENTATIVE MEASURES

Pencil and paper testing are fast becoming antiquated and displaced by the growing trend of testing inside an online environment. One of the first concerns in accessing any website is the security of the site and user verification. Usernames and passwords are used as a security verification process for accessing accounts online. Even though technology can provide students opportunities for academic dishonesty, it can also provide ways to monitor and control these opportunities (McGree, 2010). When designing an exam, instructors have the ability to use a lockdown browser. A lockdown browser, such as Respondus, will not allow students to search the Internet during the exam from the computer that they are using. This method can help eliminate cheating by making the students aware that searching for answers while taking the exam is unacceptable. The reality is that the lockdown browser can become inefficient very quick since many students use and own multiple devices such as a Smartphone or tablet which would enable them to search elsewhere.

Educators with distance learning students are confronted with a crucial issue, determining the identity of the participant. This moves to the "front and center' of the classroom experience during testing and determining the originator of written exams. Moreover, determining user identity in the virtual classroom is also linked to user progress and student aid eligibility. Student aid eligibility; of course, is linked to students meeting the minimum institutional academic requirements. To help validate student identities, keystroke dynamics are beginning to be used. Keystroke dynamics is a process that scrutinizes a user's typing style at their terminal keyboard by monitoring "keyboard inputs thousands of times per second in an attempt to identify users based on habitual typing rhythm patterns (Monrose and Rubin, 2000). "Keystroke dynamics is not what you type, but how you type" (Monrose and Ruben, 2000). Existing evidence demonstrates the reliability of keystroke rhythm to accurately determine user identity. Keystroke dynamics is inexpensive versus other biometric systems. A keyboard is the only necessary hardware that is used (Monrose and Rubin, 2000).

One way to mitigate cheating in online courses is to use a variety of assessment techniques rather than using only high stakes exams. If instructors rely on a combination of interactive discussions, writing assignments, quizzes, and projects as well as online exams, it makes it more difficult for students to cheat (Hill, 2010). Incorporating more written assignments and interaction written discussions can also help to reduce cheating. Instructors can become familiar with their students' writing styles, giving them greater confidence in recognizing possible fraudulent behavior. Plagiarism detection software can also be used on all written content as well, so the instructor has an additional method of testing authenticity of written work (Hill, 2010).

Institutions can combat cheating in a variety of ways. One method of preventing cheating is to use an "honor system" and create a culture of academic integrity within your institution. Many institutions that use an honor system require students to sign a "Pledge" either once a semester or sometimes on each examination taken, reaffirming that they are aware of the honor code and agree to abide by it. During exams, many institutions now ban all electronic devices. In addition, identification, including a photo ID or biometric scan is often now required for students taking examinations at some institutions. Pennsylvania State University's World Campus is also now testing a new online security system called WebAssessor. WebAssessor uses proctors and web cams to ensure that students match their photo IDs and allows proctors to view a student's face, keyboard, and workspace remotely. In addition, WebAssessor uses software that recognizes students' typing styles, for example, their speed of typing and whether or not they pause between typing specific letters. If the proctor sees anything suspicious, he or she can stop the student's exam immediately (Howell et al., 2010).

Some institutions require students to take exams on special "cheat-resistant" laptops. These laptops employ additional security measure and often only have software installed to display the exam. This approach has been used in Norway and also at the University of Central Florida (Howell et al., 2010).

Cabrera (2013) offers several suggestions reducing the temptation and likelihood for in "learning assessment activities, namely testing and homework activities." Tips for testing include: Purposefully select assessment methods, use question pools, randomize questions limit feedback, set timer, display questions one at a time He bases this assessment on findings from D. R. Krathwohl 2001 revision of Bloom's taxonomy. "Although it may be difficult to prevent entirely, faculty can implement steps to reduce its impact in the student learning assessment process for online courses" (Cabrera, 2013, p.1).

Cabrera (2013) offers several suggestions reducing the temptation and likelihood for "learning assessment activities, namely testing and homework activities." Tips for testing include: Purposefully select assessment methods, use question pools, randomize questions limit feedback, set timer, display questions one at a time He bases this assessment on findings from D. R. Krathwohl 2001 revision of Bloom's taxonomy.

In determining the assessment method, consideration should be given to the learning outcome sought for the student and the goals and objectives of the course. For low stake objective assessment it is suggested that purposely selected online testing questions include multiple choice, true/false, or multiple answer. Since the purpose of objective testing is to determine a student's ability to recall and organize information, other procedures can be used when accessing a student's critical thinking skills. This would include methods to determine the student's ability understand, apply, analyze, evaluate and create accurate responses to test questions (Cabrera, 2013)

Jortberg (2010) surveyed students at Sullivan University in Louisville and found that, given a choice, students would prefer to answer challenge questions in order to verify their identity before an exam rather than other identification methods such as mandatory use of a web cam, biometric scanners, signature recognition programs, or having to come to campus to take exams.

Cabrera (2013) suggests following a mix mix of objective and subjective questions. Objective measurement of understanding would involve the use of multiple choice questions, multiple answers, fill in the blank, and true / false responses. Conversely, a subjective approach would entail the use of short answers or essay questions. This subjective approach requires a greater understanding of the material. Obviously, the mixing of question types does not guarantee test questions and answers will not be shared, it could correlate in some respect to the final grade.

A tip suggested by Cabrera (2013) is to use question pools. Question pools are best used when there are a large amount of possible questions in selective categories such as true/false, multiple choice, and fill in the blank. The test administrator selects an appropriate amount of time for each question category which can vary from class to class or semester to semester. New questions can be added or old ones deleted at the inclination of the faculty member.

An option for faculty is to randomize questions. This is one of the test options available in many online learning management systems. This choice is effective since students who are administered the test at the same time are unlikely to have similar questions presented in the same sequential order. In this way, students who take the testy at the same time cannot share answers. When an option to repeat the test is permitted, this randomization provides additional security against cheating (Cabrera, 2013).

Limiting Feedback is another option in the online testing environment. Test options available in online learning management systems include, test Score, Submitted Answers, Correct Answers, and Feedback. Obviously feedback in the form of test scores must be made available to the student. If correct answers are provided or the answers submitted are marked incorrect, students' by the process of elimination can determine the correct answers and share them with others or save them for test retaking opportunities (Cabrera, 2013).Another option would be to Set a Timer. When time is used unprepared students have more to lose as looking through an open book or notes for an answer consumes valuable time. (Cabrera, 2013).

5. METHODOLOGY

This study examined faculty at two small mid-Atlantic Universities during the period of March to May 2015. The research utilized a quantitative methodology to assess faculty's implementation of authentication methods used in online courses and their experiences with ethical student behavior. The population chosen for this study was comprised of 451 faculty members at both universities. A total of 75 respondents completed the survey. The survey was conducted using Survey Monkey, an online tool, to gather and organize data. The data was imported into SPSS for further analysis. The researchers used Chi-square with a statistical significance at the .05 margin of error with a 95% confidence level. The study was a convenience sample surveying faculty from all departments within the universities which included the School of Arts and Humanities, Business, Science and Math, Engineering, Computer Science, Information Technology, Criminal Justice and Psychology. The study explores the following two research questions:

RQ1: Is there a correlation between faculty authentication methods used in online courses with age and gender?

RQ2: Do university faculty adopt techniques to mitigate the risks of unethical behavior in online courses?

The survey instrument consisted of 15 closed-ended and two open ended questions for further understanding of participant responses and feedback about unethical student behavior in online classes. The first three questions focused on faculty demographics; which included gender, age, and associated department. Questions 4 and 5 asked if the faculty members have taught online and if so, how many classes they have taught. The majority of the remaining questions aimed to ask what authentication methods the faculty used in their online courses for student identities and their experiences with students' ethical behavior. The two open-ended questions looked to understand if there were other methods that the faculty members used for validating student identities in online courses and any additional comments they may have about the topic.

6. RESULTS

Table 1: Breakdown of Participants by Age

Age Range	% of Respondents
25-30	1.35%
31-35	5.41%
36-40	25.68%
41-45	10.81%
46-50	9.46%
51-55	16.22%
56-60	16.22%
61-65	5.41%
66-70	8.11%
71-75	1.35%

The survey responses were analyzed at both universities where faculty members ranged in age from 29 years old to 72 years old. The greatest number of participants occurred in the age group of 36-40 year old representing 25.68% of the respondents with the next largest occurring between 51-55 and 56-60 years old representing 16.22% of the respondents each. The results can be seen in table 1 below. Similarly the study found that 44% of the respondents were male while 56% were female.

In order to qualify for the survey, faculty members had to teach at least one online or hybrid class. Eighty-eight percent of the 75 respondents stated they had taught online, while 12% stated they have not taught online. The respondents who stated they have taught online were asked to complete the remaining survey questions while the respondents who hadn't taught online were thanked for their participation. Participants who continued on with the survey were asked to specify how many online classes they had taught before. Approximately 83.6% of the respondents had taught less than 30 online classes with 40.98% of the total teaching less than 10 online classes. A breakdown of these results can be found in Table 2.

Table 2: Number of Online Classes Taught

Number of Classes Taught Online	% of Respondents
0-9	40.98%
10-19	24.59%
20-29	18.03%
30-39	6.56%
40-49	1.64%
50-59	3.28%
60-69	0.00%
70-79	1.64%
80-89	0.00%
90+	3.28%

The survey identified six methods that faculty could use to authenticate student identities in online courses. These methods were:

- Have students take the exam at a regional testing center or library (proctored exams)
- Used webcam proctors
- Used keystroke recognition devices
- Used independent coursework instead of multiple choice or true/false exams
- Utilized Respondus lockdown browser
- Utilized Turnitin for assignments, projects and papers

Each participant was permitted to select as many answers that applied to their teaching methods. The most utilized methods included using Turnitin for assignments and independent coursework at 52.31% and 44.62 %, respectively. None of the participants said they used keystroke recognition devices. These results can be reviewed in Table 3.

Table 3: Student Identity Authentication Methods

Method Used	% **of Respondents**
Proctored Exam	6.15%
Webcam Proctor	3.08%
Keystroke Recognition Devices	0.00%
Independent Coursework	44.62%
Respondus Lockdown	21.54%
Turnitin	52.31%

The researchers wanted to understand if any of the six authentication methods were statistically significant when compared with either Gender or Age. Proctored Exams were the only authentication method that was statistically significant with gender. Additionally, using Turnitin was slightly outside of this range with a chi-square value of .064. Age was statistically significant with Proctored Exams, while none of the other methods had a chi-square value within the acceptance criteria. It is important to note that using keystroke recognition devices did not have any responses and therefore are blank in the table below. A full breakdown of these results can be seen in Table 4 in the appendix.

The researchers wanted to better understand what faculty members do to prevent online cheating and what their level of awareness was in relation to websites that aide students with unethical behavior in online courses. The results of these questions are found in Tables 5 and 6 in the appendix and correlate to the following questions:

- Require students to sign an academic integrity policy for their online class.
- Aware of online services where students pay someone to take online courses for them.
- Give the same Exam each semester for online classes.
- Checked online sites where students past copies of online exams.
- Require students to read the university’s code of conduct.
- Believe that Students cheat more in online classes than in the classroom.
- Caught students turning in work that wasn’t theirs.
- Caught Student turning in work from a student in a previous class.
- Required the use of a proctor or administer for an exam.

7. DISCUSSION

The first research question attempted to discover if a correlation existed between age and gender with the authentication methods used. With such a wide-spread age group and a fairly distrusted gender population, the researchers hoped to analyze any trends or correlation that existed. Proctored exams were statistically significant with both age and gender, having a chi-square value of .034 and .02, respectively. Based upon these results, it seemed as age increased, faculty relied more heavily on proctors for their exams. It is possible that the younger generation of faculty are implementing other technological solutions for their exams or potentially modifying their assignments to promote independent work and analysis of the subject matter.

Although, it did not meet the requirements of being statistically significant within a value of .05, the use of Respondus Lockdown technology in relation to Age had a chi-square value of .097. Additionally the use of Turnitin.com for assignments analyzed with age had a chi-square value of .064, which was fairly close to the predefined boundary of .05 utilized to determine statistical significance. While none of the remaining authentication methods showed much of a statistical significance, the researchers felt it was important to note these two authentication methods given their chi-square values. It is possible that age had somewhat of an impact on the use of technology like Respondus and gender had an impact on the use of sites like turnitin.com.

Our second research question focused on preventative measures faculty used to prevent cheating and any experience they may have encountered with students conducting unethical behavior in an online course. Overall, 58.46% of the faculty felt that students cheat more in online classes than they did in traditional courses. Additional comments we received about this topic included faculty suggesting that students had more liberty to use external resources to aid in their exams and assignments than the opportunities available during a traditional course. Supplementing this information, 82.81% of faculty stated they have caught students turning in work that wasn't theirs for an online class. Forty percent of the participants caught students submitting work in an online class from a student who previously took the class. With these results being much higher than expected, the researchers were interested to understand if faculty were aware that services exist to help students cheat in online courses and are faculty actively working to combat this unethical behavior. One of the faculty members responded that they had a student who posted an ad on Craigslist asking someone to help him complete his coursework and exams. After some altercation between the student and the person who responded to the ad, the professor was contacted by the person who responded to the ad. He described the interactions with the student, the arrangements made, and provided detailed email communication illustrating the student was cheating in the course.

The researchers were a little surprised by the results of the survey questions related to the faculty members' efforts to combat unethical online behavior. Only 21.54% of the respondents had students sign an academic integrity policy. Additionally 40% of the faculty found examples where students submitted work from other students in previous semesters. Of the respondents, only 9.23% stated they reviewed online sites to see if copies of their exams and assignments were posted. At the end of the survey, the participants had the opportunity to submit comments based upon their responses. A large number of comments entailed faculty stating they were unaware of these sites and services and stated they will be making a better effort to implement the use of an integrity policy in their online courses. Given the lack of familiarity with these services and the participant feedback, the researchers concluded that with additional awareness, faculty may be more inclined to safeguard their courses against these unethical student acts.

8. CONCLUSIONS

Academic integrity has posed challenges to educators for many years. While universities continue to adopt technological advancements that offer a less restrictive environment for students, they also pose an additional layer of risk in identifying student work and student identities for online courses. Many will say the first step to mitigate these risks is awareness of techniques students could adopt for unethical behavior in online courses. Based on feedback from this survey, the researchers are confident that many faculty would benefit from awareness and are willing to take the extra step in actively preventing unethical student behavior in their online courses.

However, another element to this issue is that certain faculty find teaching online "convenient" and merely try to replicate there in-class teaching methodology into their online course. For these faculty, awareness isn't the biggest hurdle they face. Some administrators might believe that the best solution is to merely prevent these faculty members from teaching online if they are unwilling to put forth the extra effort to prevent cheating. Another option is for the university to invest in additional staff or resources that can help aid the faculty in this initiative. Universities have many options available to them from various sites that will help assess the level of originality in student papers to services that offer proctoring of exams to ensure student identities. Of course, each of these options comes at a cost that universities may not be willing to absorb given the economic stresses surrounding higher education.

However, failure to take preventative measures will likely cause negative effects to the university's reputation along with a potential decrease in enrollment if the university is perceived of lesser quality to their competitors by not implementing authentication measures in the online classroom.

9. REFERENCES

Baile, J.L, & Jorbert, M.A. (2008). Distance learning student authentication: verifying the identity of online students. *Finding Common Ground : Programs, Strategies and Structures to Support Student Success,* Vol. 3.

Cabrera, D. (2013, March). Tips to reduce the impact of cheating in online assessment. Northern Illinois University Instructional Design Center. Retrieved on May 17, 2015 fromhttp://facdevblog.niu.edu/onlinecheating

Database of Federal Legislation. (2007). Higher education opportunity act. Retrieved on June 5, 2015 from https://www.govtrack.us/contress/bills/110/hr4137

Grijalva, T., Kerkvliet, J., & Nowell, C. (2002). Academic dishonesty and online courses. Retrieved April 10, 2015 from http://oregonstate.edu/dept/econ/pdf/cheat.online.pap6.pdf

Hill, C. (2010). Student authentication: What are your duties under the HEA reauthorization? *Faculty Focus Special Report: Promoting Academic Integrity in Online Education.* 4-7.

Howell, S., Sorenson, D., & Tippets, H. (2010). The news about cheating for distance educators. *Faculty Focus Specialty Report: Promoting Academic Integrity in Online Education,* 4-7.

Jortberg, M. (2010). Experiencing verifying the identity of distance learning students. 26th Annual Conference on Distance Teaching and Learning. pp. 1-5.

Krask, A.M. (2007). Curbing academic dishonesty in online classes. TCC 2007 Proceedings. Retrieved on May 20, 2015 from http://etec.hawaii.edu/proceedings/2007.krsak.pdf

Krathwohl, D.R. (2001). A revision of Bloom's taxonomy. An overview in L.W. Anderson and D.W. Krathwohl Eds, A taxonomy for learning, teaching and assessing. A revision of Blooms Taxonomy of Educational Objectives, New York, Longman

McGee, P. (2010). Supporting academic dishonesty in online courses. TCC 2007 Proceedings. Retrieved on June 2, 2015 from http://www.thejeo.com//Archives/Volume10Number1/McGee.pdf

Michael, T.B., & Williams,, M.A. (2013). Student equity: Discouraging cheating in online courses. *Administrative Issues Journal, Education, Practice and Research.* Retrieved on May 1, 2015 from http://dx.doi.org/10.5929/2013.3.2.8

Monrose, F, & Rubin, A.R. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems,* 16 (2000), 351-359

Moten, J., Fitterer, A., Brazier, E., Leonard, J., & Brown, A. (2013). Examining online college cyber cheating methods and prevention measures. *Electronic Journal of e-Learning, 11(2),* 139-146.

Paullet, K., Douglas, D.M. & Chawdhry, A. (2014). Verifying user identities in distance learning education. Who is sitting and submitting from behind the screen? *Issues in Information Systems,* Vol 15, Iss, 1, 370-379

Rogers, C.F. (2006). Faculty perceptions about e-cheating during online testing. *Journal of Computing Sciences in Colleges,* 22(2), 2060212

Rowe, N.C. (2004). Cheating in online student assessment. Beyond plagiarism. *Online Journal of Distance Learning Administration* 7(2). Retrieved May 1, 2015 from http://www.west.ga.edu%7edistance/ojdla/summer72/rowe72.html

Watson, G., & Sottile, J. (2010). Cheating in the digital age. Do students cheat more in online classes? *Online Journal of Distance Learning Administration.* Retrieved May 17, 2015 from http://www.west.ga.edu/distance/ojdla/spring131/watson131.html

Appendices and Annexures

Table 4: Chi-Square Analysis of Authentication Methods

Chi-Square Analysis		Gender			Age	
Proctored Exam	5.378	1	0.02	48.12	32	0.034
Webcam Proctor	1.614	1	0.204	75	32	0
Keystroke Recognition Devices	---	---	---	---	---	---
Independent Coursework	0.132	1	0.717	20.159	32	0.948
Respondus Lockdown	0.48	1	0.489	42.733	32	0.097
Turnitin	3.424	1	0.064	31.598	32	0.487

Table 5:

	Integrity Policy	**Online Services**	**Same Exams Each Semester**	**Online Sites**
No	78.46%	60.00%	81.54%	90.77%
Yes	21.54%	40.00%	18.46%	9.23%

Table 6:

	Code of Conduct	**Cheating Online**	**Other Student's Work**	**Previous Work**	**Using a Proctor**
No	47.62%	41.54%	17.19%	60.00%	89.06%
Yes	52.38%	58.46%	82.81%	40.00%	10.94%

Game Development as a Pathway to Information Technology Literacy

Mark Frydenberg
mfrydenberg@bentley.edu
Computer Information Systems Department
Bentley University,
Waltham, MA 02452, USA

Abstract

Teaching game development has become an accepted methodology for introducing programming concepts and capturing the interest of beginning computer science and information technology (IT) students. This study, conducted over three consecutive semesters, explores game development using a gaming engine, rather than a traditional programming language, as a means not only to introduce programming concepts, but also to promote the development of information and communications technology (ICT) literacy skills among first-year business students. The paper argues that in addition to learning programming concepts, completing the steps involved to develop and publish an original game requires students to demonstrate a variety of ICT skills. To be successful, they must be proficient at creating and editing multimedia, interacting with multiple operating systems and mobile devices, performing research online, transferring files from one machine to another, and uploading the files for their games to an app store and the web.

Keywords: game development, ICT skills, information technology literacy, app development, programming.

1. INTRODUCTION

Teaching technology literacy to non-technical students has changed greatly since John Kemeny and Thomas Kurtz, creators of the BASIC programing language at Dartmouth College, introduced the notion of computer literacy in 1964. Kemeny recognized that that "someday computer literacy will be a condition for employment, possibly for survival, because the computer illiterate will be cut off from most sources of information." (Kemeny, 1983) Computer literacy then was achieved by using time-shared teletype terminals connected to a college mainframe to create simple BASIC programs as a way to develop algorithmic thinking. "By making the BASIC environment so friendly, [Kemeny and Kurtz] created a safe place for people to play and explore. The computer game movement came from BASIC. People shared their games, long before there were networks, by printing the programs [in computer magazines] for others to enter in and enjoy." (Claburn, 2014)

Today, the computing tools, languages, and devices have evolved, but creating games as a pathway to achieving technology literacy has remained constant. The author's previous research (Frydenberg, 2015) examines how game development provides an authentic learning experience. This paper examines both programming and general information and communication technology (ICT) concepts that students may learn as a result of developing and publishing their own computer games to an online app store and the web.

Computer literacy has expanded to include digital literacy and ICT literacy. The ability to write programs is only one possible outcome of technology literacy today. Equally, if not more important than knowing how to program is for today's students to have proficiency with personal productivity and collaboration tools, email, the World Wide Web, social media, mobile devices and the cloud, and to know when and how to use these tools responsibly. (Frydenberg & Press, 2009) (Press, 2011) "Digital literacy comprises effective problem-solving skills, critical thinking and communication skills, creativity, and self-regulation along with an understanding of culturally and contextually based practice in the use of, and engagement with digital technologies." (Burton, Summers, Lawrence, Noble, & Gibbings, 2015, p. 152)

The Partnership for 21st Century Living (2015) describes ICT literacy as the ability to apply technology effectively, including:

- Use technology as a tool to research, organize, evaluate and communicate information
- Use digital technologies (computers, PDAs, media players, GPS, etc.), communication/networking tools and social networks appropriately to access, manage, integrate, evaluate and create information to successfully function in a knowledge economy
- Apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

Game Development and Technology Literacy

A current trend in both high school and college introductory computing courses is to incorporate the use of visual development tools to enable students to build apps and games with little or no programming required. (Catete, Peddycord III, & Barnes, 2015), (Abelson, Mustafaraj, Turbak, Morelli, & Uche, 2012) (Thomas & Blackwood, 2010). Popular game development tools include Alice (Carnegie Melon University, 2015), Scratch (Lifelong Kindergarten Group at the MIT Media Lab, 2015), App Inventor (MIT, 2015), and Game Maker (YoYo Games Ltd., 2015). One reason for this interest among faculty is to "better attract and retain students in the computer science major." (Bayliss, 2009, p. 337)

From an information technology educational perspective, creating games presents students with opportunities to be creative, and demonstrate their proficiency as able participants in a world based on technology literacy skills. Games allow students to turn their ideas into real applications that they can play on their own laptops and mobile devices.

While game development in and of itself is a worthwhile exercise for students studying programming, this paper argues that information technology and business students also can develop additional ICT literacy skills and understand IT concepts through the process of creating computer games.

Research Questions

This study shares results after offering a game development experience to beginning information technology students. The study addresses the following research questions:

- How does game development fit within the topics of an introductory information technology course?
- Which ICT skills do students find important while creating computer games?
- Does creating and publishing a computer game allow students to apply their knowledge of IT concepts and develop ICT skills?

2. LEARNING IT CONCEPTS THROUGH GAME DEVELOPMENT WITH CONSTRUCT 2

This study describes the use of Construct 2 (Scirra Ltd., 2015), a game development tool for creating two-dimensional games, among first-year business students in an information technology course. Construct 2 does not require learning a programming language and allows for rapid prototyping of computer games, so it is quite suitable for beginners to create a functional game with no prior programming skills. Construct 2 includes a development environment to design a game and specify its rules of play, and a Game Creator to export completed games as web or mobile apps. Construct 2 was chosen for this study because its Game Creator generates HTML5, CSS3, and JavaScript files that can be published on the web, as well as native app packages for Windows Phone, iOS, and Android, that can be published on the Windows, Apple, and Google Play app stores and marketplaces.

The sections that follow discuss the programming concepts to which students are exposed while developing their games, and the IT skills that they develop while creating the game's multimedia elements, and publishing their games on the web and to an online app store.

Learning Programming Concepts through Game Development

To develop a game in Construct 2, one begins by designing the game's screens, layouts, and objects (characters, platforms, projectiles, sounds, and so on), and their properties. Figure 6 shows the Krazy Kopter game during play. The game was created by a student in this study. The object of the game is to fly the helicopter without hitting an upper, lower, or middle obstacle. The heights of the obstacles are generated randomly.

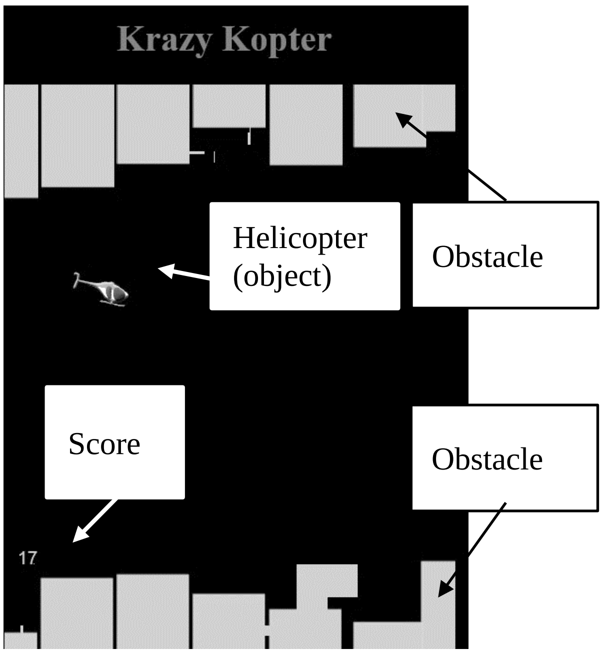


Figure 6. The object of Krazy Kopter is to fly the helicopter and avoid the obstacles.

Appendix 1 Figure 1 shows the game's start screen, and the how the game's play layout is designed using Construct 2. The developer can assign properties to each object in the game.

Expressing objects and their properties in terms of real world objects, such as a helicopter's position or speed, is a natural way to introduce the concept of objects. As with courses introducing objects through Alice and Scratch, Construct 2's environment allows students to develop "a good sense of objects." As Stephen, Dann, and Pausch write, "Everything in a student's virtual world is an object!" (Stephen, Dann, & Pausch, 2003, p. 193). The main characters of their games, whether flying helicopters or flapping birds, are all objects. Figure 7 shows some of the properties of the helicopter object.

Construct 2 uses a visual syntax to describe events that occur during game play (such as screen taps, key presses, and collisions) and the actions to take in response to them. Identifying these steps writing the instructions introduce students to basic input/processing/output and promote algorithmic thinking. Students learn that they must initialize variables at the start of the game (such as setting constant values for speeds or starting values for the game score), and describe the steps to follow when events are detected at each iteration of the play loop, or "tick" of a timer, that continues until the game is over.

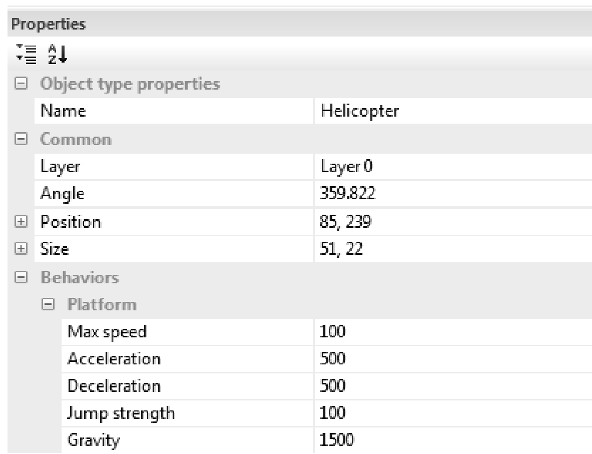


Figure 7. Expressing properties in Construct 2.

Students find that expressing rules in natural language is often helpful before constructing them using Construct 2. For example, one could summarize the conditions for ending the game as "When the helicopter collides with an upper, lower, or middle object, return to the Start screen (because the game is over)." Figure 3 shows how to express this rule in Construct 2.

Figure 8. Specifying the end-of-game rule in Construct 2.

Figure 2 in Appendix 1 shows all of the rules for the Krazy Kopter game, annotated with brief comments describing what they do. Table 4 below summarizes the programming concepts that are used the Krazy Kopter game design. Specifying the game play in Construct 2 introduces students to fundamental programming constructs, including creating instances of objects, constants, program logic, and transfer of control.

Table 4. Programming Concepts Reflected in using Construct 2 to Design the Game shown in Appendix 1 Figure 2

Step	Programming Concept Reflected
0	Initialize constants and variables.
1	Read input (from touch or keyboard)
2	Looping and iteration
3	Create new instances of an object
4	Create new instance of an object
5	Decision logic
6	Destroy objects no longer in use
7	Logical comparison, increment variables, display output

Learning Information and Communications Technology Skills and Concepts through Game Development

In addition to learning programming concepts, students apply their ICT knowledge to this project as they interact with a variety of multimedia content and apps to complete their games. By using Construct 2 to create a game that runs as a web app or a Windows app, students experience the prevalent trend in software development today of creating apps that run on multiple devices and platforms. (de Andrade, Albuquerque, Frota, Silviera, & da Silva, 2015) Students see the relevance of creating software that runs on their computer or mobile device.

Through the process of making games, students demonstrate proficiency with and understanding of several technology-related tasks and concepts.

3. IMPLEMENTATION OF STUDY

The game development study described in this paper took place over three semesters (spring 2014, fall 2014, and spring 2015), in three different sections of a technology concepts course at Bentley University, a business university in the United States. IT 101 (Introduction to Information Technology and Computing Concepts) is a required course for all first year students. IT 101 teaches technology literacy skills including maintaining laptops, using mobile and desktop productivity software and apps, navigating the World Wide Web, developing simple web sites with HTML, interacting with operating systems. The course also introduces IT concepts including storing and communicating digital information, multimedia formats, Internet protocols, digital storage and the cloud, computer and networking components, mobile devices, and building technology solutions.

Table 5. ICT Concepts and Skills Evident in Game Design and Publishing Tasks.

Development Task	ICT Concept / Skill
Design and develop a 2-D game using Construct 2.	Programming logic and languages, objects, instances, variables, transfer of control, iteration
Create or locate sound and image files for use in the game.	Multimedia files, downloading, resizing, file compression, Creative Commons, Online tools, search.
Export the game from Construct 2 for web and Windows Store.	Files and Directories, Compile.
Publish HTML version of game on website.	HTML5, create pages with <embed> code, FTP files to a web server
Play game on smartphone or tablet.	Native vs web apps; Shortcuts.
Copy game files from user's (Windows 7) laptop to a Windows 8 computer to submit to Windows Store.	Copy files from one device to another. Flash memory, cloud storage. Interact with multiple operating systems.
Prepare images for game's logo, store logo, and splash screen.	Basic image editing: resize, crop
Digitally sign files for submission to the Windows Store.	Digital Signatures
Test game in emulator as part of submission process to Microsoft store.	Emulator, Software deployment.
Respond to errors in certification reports if game does not pass submission.	Software development life cycle, debugging, software revisions

In-Class Training

Students enrolled in three different sections of the course taught by two different instructors participated in the project each semester:

- An Honors section, for students enrolled in the honors program, chosen because of their high scholastic abilities,
- An Accelerated section, offered to students who self-selected to be this section because of their interest in technology, and
- A standard section open to all students.

Each section met for two 75-minute sessions per week.

Students installed Construct 2 on their laptops prior to the first training session. At the first training, they followed a demonstration to create a simple 2-D platform game. Completing this exercise introduced game elements such players, shooters, and layouts, user interface elements for keyboard and touch input, detecting collisions of objects, and creating and destroying objects. During the second training session, students learned to enhance their games with music and sounds, keeping score, and end-of-game logic.

Microsoft sponsored the gaming project and provided support from an academic technology evangelist to lead the training sessions in each class, offering consistent delivery across all sections. The instructors, student tutors, and former IT 101 student volunteers who had completed the gaming contest in prior semesters, assisted students during the demonstrations and exercises in their classrooms. Student tutors and the Microsoft evangelist provided additional support outside of class during designated hours in the CIS Sandbox, a campus tutoring and social learning space.

As an additional incentive, Microsoft donated prizes to award to the creators of the best games at a games party held at the conclusion of the project.

Student Assignment

The gaming project spans a period of four weeks during the semester, so students have time to gain basic proficiency with Construct 2, design their own games, and get lots of help from tutors and peers. This is one project where the process of completing it is as important as the final result, if not more, when determining a grade. Dividing the project into smaller steps seems to make the process more manageable.

The gaming assignment required students to create an original game, or adapt the game they worked on in class and add new features. They could collaborate with a partner in the design process, but had to create their own individual games (which could be similar in game play) for this assignment. Students had to submit their games to the Windows Store, and publish them to their websites created earlier in the semester. Appendix 2 Table 1 lists several online resources shared with students for use in completing their assignment. The complete description of the assignment is provided in Appendix 3.

Tutors and former IT 101 students played and reviewed all of the games the weekend after they were due, evaluating them based on their originality, playability, complexity, and aesthetics. The top seven games were selected from these reviews. All students then were invited to a games party where they could play all of the games and vote for their favorites. The winners were announced, and prizes awarded.

4. STUDENT SURVEY

Approximately 269 students enrolled in 9 sections of IT 101 over a three-semester period participated in this study. At the end of each of the three semesters, a survey was offered to each class participating class. Responses were received from 239 students; 64% were male, 36% female. Not all students replied to all questions used in this study.

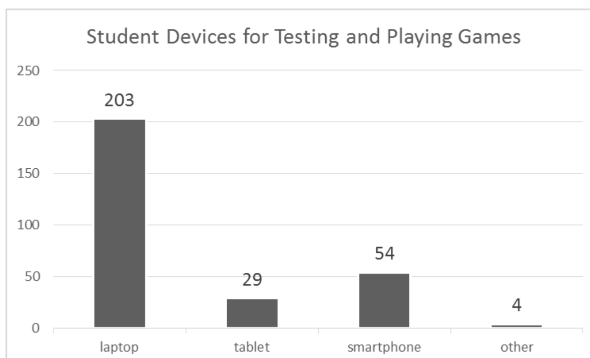


Figure 9. Devices used for testing and playing games.

When asked if they created any type of software applications in the past, 85% of the 208 responses indicate that they had not; 15% had created applications before, including simple games, web pages, Java, or C++ programs. These results suggest that creating games to play on their computers and mobile devices was a new experience for most of the students, and certainly, deploying them to two platforms was new for all of the students. Figure 9 shows the devices students used to test or play their games. It is no surprise that 98% used laptops (since each student has one), followed by smartphones (26%), tablets (14%), and other devices (4%).

Appendix 1 Figure 3 shows attitudes toward technology among the students enrolled. As enumerated in Table 2 above, creating a game allows students to experience many IT concepts beyond programming, and demonstrate ICT skills as they complete it.

Figure 5 summarizes student perception of the importance of several ICT skills. Students used a scale from 1 (not important) to 5 (very important) as they rated the importance of each to the gaming assignment. Values shown are the average ratings for each skill.

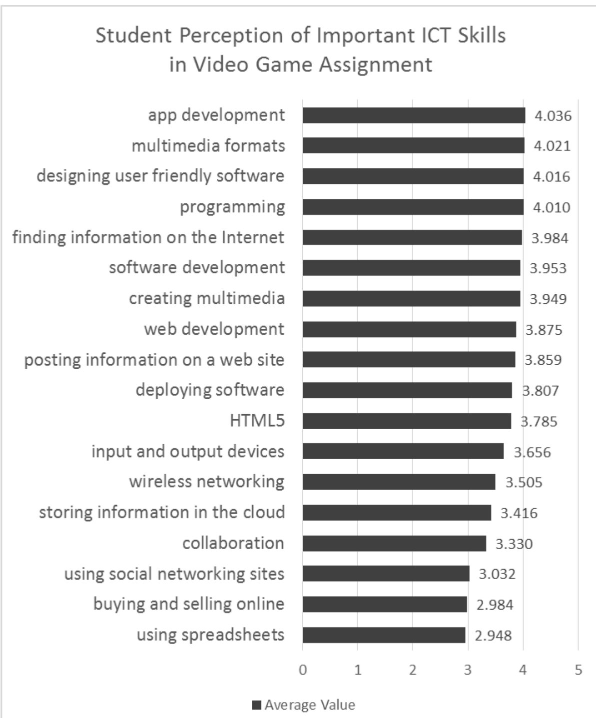


Figure 5. Student perception of important ICT Skills.

Students recognized that creating multimedia, knowledge of multimedia formats, programming, software, app, and web development, HTML5, deploying software, and designing user friendly software as more important to creating their games. These were skills they had to apply while completing their games. Students found the remaining ICT skills to be less applicable to this project.

Appendix 1 Figure 4 summarizes how students experienced the game project as a way to apply their IT knowledge and skills. 78% of the 208 students responding to this question students agreed or strongly agreed that the project enabled them to learn about app development; 55% agreed or strongly agreed it helped them develop skills in logical thinking; 41% agreed or strongly agreed it let them use IT 101 skills; 42% agreed or strongly agreed that it captured their interest in CIS; 63% agreed or strongly agreed that the gaming assignment was an interesting way to apply IT concepts. These results suggest that the project was an effective way for students to develop ICT skills and apply their IT knowledge.

Said one student: "This project introduces very important programming concepts to IT 101 students. It may not be for everyone, especially those who are not interested in developing applications. " Another student commented, "It is obvious how [creating a game] developed my skills working with multimedia as that is how I added graphics and such, but it helped me understand how a digital game interacts with a website and then both of those interact with a server, when I posted it. A very helpful lesson!" A third student commented on the business skills required. "The main thing I kept running into was problem-solving and collaboration with others. These are two very helpful real-world skills, so it was good that they were integrated in the project."

5. CHALLENGES AND LESSONS LEARNED

The process of deploying a game to two different platforms (the web and the Windows Store) was a challenge for some students.

Publishing Games to the Web

Most students were able to follow the instructions provided to export their games from Construct 2 for deployment to the web. Students added pages to their web sites to include screenshots and descriptions of their games, the embed code provided from Construct 2, and a link to a web page for playing the game full screen.

To simplify the process of publishing games to their websites, students were instructed to save the HTML5, JavaScript, CSS3, and other files generated by the Game Creator in a Game folder within their website. Some students forgot this step, and saw that their website's index.html file was unexpectedly overwritten with the game's index.html file because it, and the other game files were not saved in the correct location.

Once published to their websites, students could navigate to their game's web address using a browser on their laptop or mobile device to play the game. By adding this location as a shortcut on their devices' home screens, their games became original, fully-functioning mobile web apps. Students were able to "experience mobile computing as creators, not just consumers. This goes beyond the issues of teaching programming or computational thinking, to fundamental ideas about how a citizenry can be empowered in the age of information." (Abelson, Mustafaraj, Turbak, Morelli, & Uche, 2012, p. 39)

Publishing to the Windows Store

Students also exported their Construct 2 games as native apps to submit to the Windows Store. While most students did not have their own Windows 8 laptops or mobile devices on which to run their games as Windows 8 apps if published successfully, the process of submitting games to the Windows Store still has great educational value.

Games must pass strict tests in order to be published in any vendor's app store. The submission process verifies that the correct files are present, that the game or app runs without crashing, and that the submission contains the required documentation. When submitting to the Windows Store, Microsoft testers play each game, testing for touch input and other features.

Publishing a game to the Windows Store was by far the biggest challenge to students each time the project was offered. The hardest part for many students turned out to be following the multi-step instructions provided. Even though screen-by-screen instructions showed what to do in Visual Studio, many students were frustrated quickly. This is likely because they students had no previous familiarity with Visual Studio and because they had to user computer lab machines running Windows 8 to do so. Most student laptops and all university computer labs were still running Windows 7, but the version of Visual Studio required for submitting their games to the Windows Store ran on Windows 8. Windows 8 was installed on four computers in the lab, but that still was not sufficient. The instructors increased this to eight computers the following semester, which was an improvement, but still caused long waiting lines to access a computer from which students could submit their games to the Windows Store.

Once their games were accepted, however, many students took pride in their accomplishments, some noting that it is a resume booster to have a game of their own creation published to the Windows Store.

In some cases, interpreting comments on a report if a game did not pass certification also was challenging. For example, a content and age rating must be specified at the time of submission, but a student neglected to do so. There was nothing wrong with the game itself, but the information needed for publication was incomplete.

From an IT literacy point of view, following the process of publishing an app to an online store gives students a better appreciation of the steps that developers of the apps they use every day must follow, and adds a real-world learning lesson, beyond programming skills, to the project.

Microsoft requires a free Windows developer account to submit games to the store. Credit card verification was required for identifying a user's country of origin in order to submit a game to the store. Not all students had their own credit cards, which added a level of complexity. Purchasing a few pre-paid Visa cards was an awkward, but sometimes successful workaround. Microsoft has since removed this requirement.

In subsequent semesters, the instructors created one "class account" for each section for students to use when submitting their apps. Using a shared account for the entire class simplified the instructors' task of tracking the status of the games that students submitted. By signing to the Windows Dev Center at http://dev.windows.com with the class credentials, the developer's dashboard shows the current status of all apps associated with the account, as shown in Figure 6.

Figure 6. The Windows Dev Center shows the status of submitted apps.

Microsoft's lifting the credit card requirement combined with the shared class account dramatically increased the number of students who were able to publish their games on the Windows Store successfully, as shown in Figure 7. The first time through, the instructors modified the assignment's publication to the Windows Store requirement to become extra credit because it was such a burden to do. By the third iteration of the project, attempting to publish to the store was required.

Figure 7 summarizes the percentage of students each semester who reported that their games were published successfully during each semester of this study, along with those who tried and gave up, and the small number who didn't bother to try.

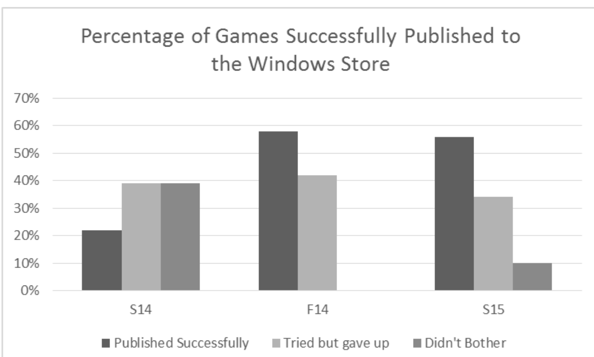


Figure 7. Games successfully published to the Windows Store.

Student Attitudes

Despite the fact that many students found creating their games to be challenging, and publishing it to the Microsoft Store to be frustrating and overwhelming, students took pride in the games they created. Over 100 of the 130 students who responded affirmatively to the question "Did you show your game to anyone else?" also added comments, such as "they thought it was cool that I could do that…," "they were impressed," or "my friends were amazed that my IT class was making games." Students worked hard at the assignment, 50% spending more than 6 hours on it, as shown in Figure 8.

Said one student, "While I created my game, I gained knowledge in the software development process and understood how game designers built their own games."

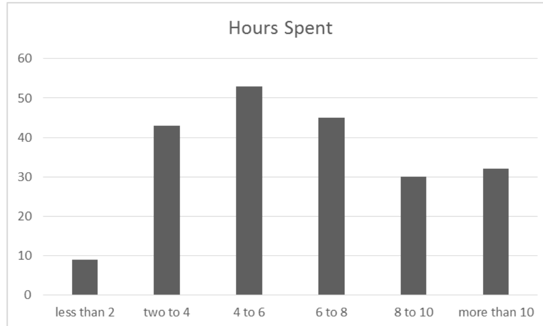


Figure 8. Hours Spent on the Game Assignment.

6. OBSERVATIONS AND CONCLUSIONS

While students were required to create their own games, they were encouraged to work in pairs to share ideas and review each other's work. This proved to be effective as students were able to support each other in their learning and show each other how to accomplish various tasks they remembered but their partners did not. Students used appropriate online ICT tools to share screens, graphics, or other files with each other.

The games that students created were completely functioning apps that they could run on their computers and mobile devices. Many featured touch input, music, score keeping, animations, and other characteristics found in real games. Not only did students create games, they published their games on two different platforms so others could access them on their devices. Doing so required aptitude to identify and solve basic tasks on their computers and devices. All students, regardless of their academic designation as characterized by the section of IT 101 in which they were enrolled, completed at least a basic game.

Completing the steps to publish a game successfully on the Windows Store required a level of support from student tutors, faculty, and the Microsoft evangelists that was not sustainable, and led to frustration among several students. During a later semester, the instructors replaced the requirement of publishing games as native apps to the Windows Store with the simpler task of publishing them as web apps to the Microsoft Azure cloud platform. By publishing to the cloud, students had to configure a virtual FTP server, make use of cloud services, and interact with the Azure portal. In some cases, students saw how the geographic location of the server on which their game was stored impacted performance. They learned that the consumer cloud storage tools with which they were familiar, such as Google Drive, Microsoft OneDrive, and Dropbox, were useful to store and publish files to the web, but a different solution was needed to publish applications to the web.

Quality of games was mixed across all sections. While all students completed the same training, the games they created were very different. Even variations on the same in-class example generated a variety of results and outcomes. Many students adapted online tutorials to create simple adventure, platform, shooter, and other genres of 2D games. A games contest offered added incentive to produce a high quality final product.

7. ACKNOWLEDGEMENTS

The author acknowledges Michael Cummings and Gavin Baumann, Microsoft Academic Evangelists for their continued dedication to this project. Thanks to Bentley University student Brandon O'Connell for sharing screenshots of his Krazy Kopter game.

8. REFERENCES

Abelson, H., Mustafaraj, E., Turbak, F., Morelli, R., & Uche, C. (2012). Lessons learned from teaching App Inventor. *Consortium for Computing Sciences in Coleges, 27*(6), 39-41.

Bayliss, J. D. (2009). Using games in introductory courses: tips frm the trenches. *41*(1), 337-341.

Burton, L. J., Summers, J., Lawrence, J., Noble, K., & Gibbings, P. (2015). Digital Literacy in Higher Education: The Rhetoric and the Reality. In M. K. Harmes, *Myths in Education, Learning and Teaching: Policies, Practices and Principles* (pp. 151-172). Pallgrave Macmillan.

Carnegie Melon University. (2015). *Alice*. Retrieved from http://www.alice.org/index.php

Catete, V., Peddycord III, B., & Barnes, T. (2015). Augmenting introductory Computer Science Classes with GameMaker and Mobile Apps. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 707-709). New York, NY: ACM.

Claburn, T. (2014, 04 30). *When BASIC Was Young: Great Memories*. Retrieved from http://www.informationweek.com/strategic-cio/executive-insights-and-innovation/when-basic-was-young-great-memories/d/d-id/1234939

de Andrade, P. R., Albuquerque, A. B., Frota, O. F., Silviera, R. V., & da Silva, F. A. (2015). Cross platform app: a comparative study. *International Journal of Computer Science & Information Technology, 7*(1), 33-40.

Frydenberg, M. (2015). Creating Games as Authentic Learning in the Information Technology Classroom. *Proceedings of the International Conference E-Learning 2015* (pp. 113-120). Gran Canaria: IADIS.

Frydenberg, M., & Press, L. (2009). From Computer Literacy to Web 2.0 Literacy: Teaching and Learning Information Technology Concepts Using Web 2.0 Tools. *Infomation Systems Education Journal, 8*(10), 18.

Kemeny, J. G. (1983). The Case for Computer Literacy. *Daedalus, 112*(2), 211-230.

Lifelong Kindergarten Group at the MIT Media Lab. (2015). *Scratch*. Retrieved from https://scratch.mit.edu/

MIT. (2015). *MIT App Inventor*. Retrieved from http://appinventor.mit.edu/explore/

Partnership for 21st Century Living. (2015). *Framework for 21st Century Living*. Retrieved June 1, 2015, from Partnership for 21st Century Living: http://www.p21.org/about-us/p21-framework/350-ict-literacy

Press, L. (2011, October 24). *IT literacy – evolution, curriculum and a modular e-text.* Retrieved March 3, 2015, from http://som.csudh.edu/fac/lpress/presentations/mahyar.docx

Scirra Ltd. (2015, 06 01). *Construct 2*. Retrieved from Scirra: https://www.scirra.com/construct2

Stephen, C., Dann, W., & Pausch, R. (2003, January). Teaching Objects-first in Introductory Computer Science. *ACM SIGCSSE Bulletin, 35*(1), 191-195.

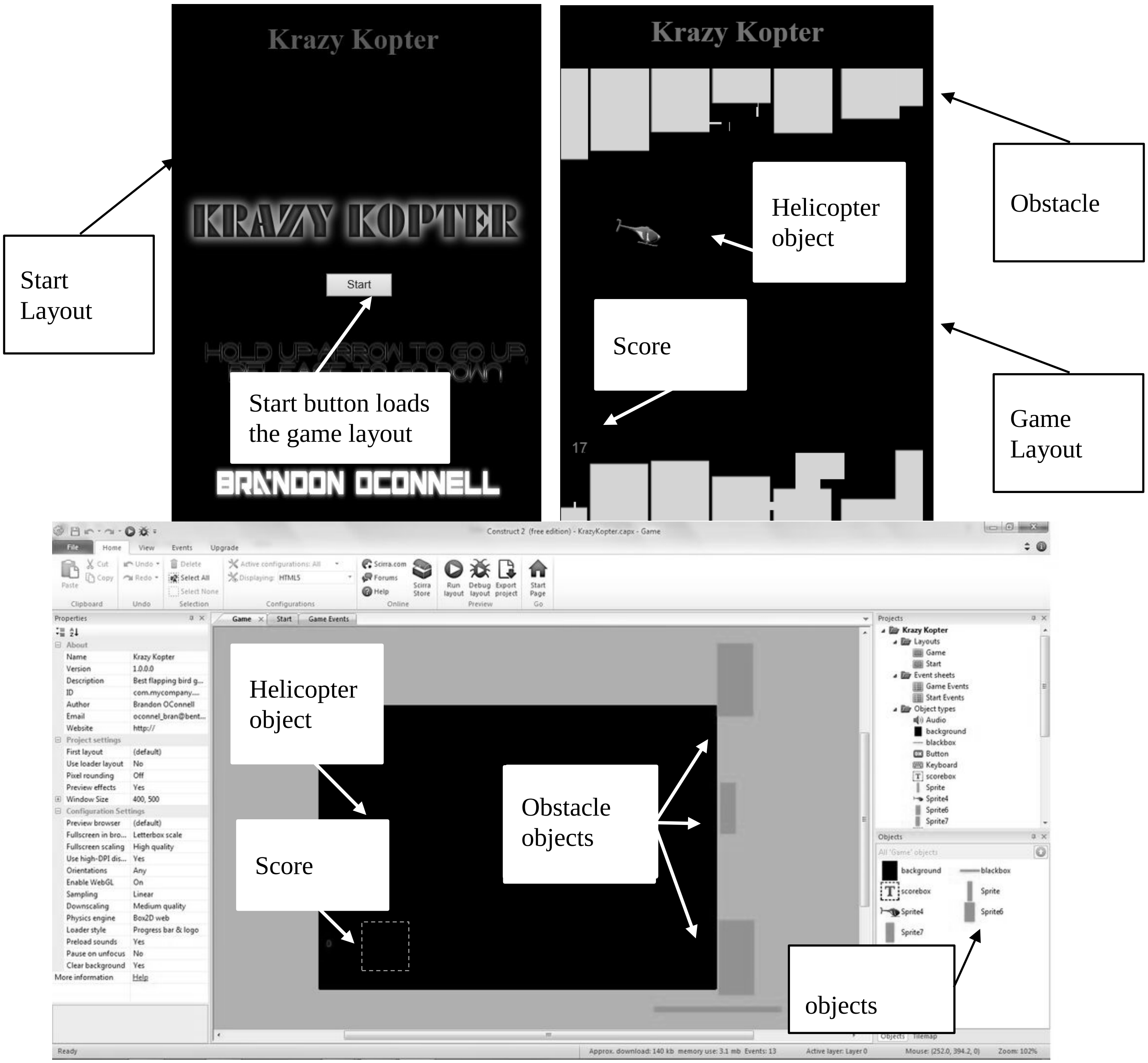
Thomas, J. D., & Blackwood, M. (2010). Computer Literacy and Non-IS majors. *Information Systems Education Journal, 8*(58), 3-12. Retrieved from http://isedj.org/8/58

YoYo Games Ltd. (2015). *GameMaker: Studio*. Retrieved from Yo Yo Games: http://www.yoyogames.com/studio

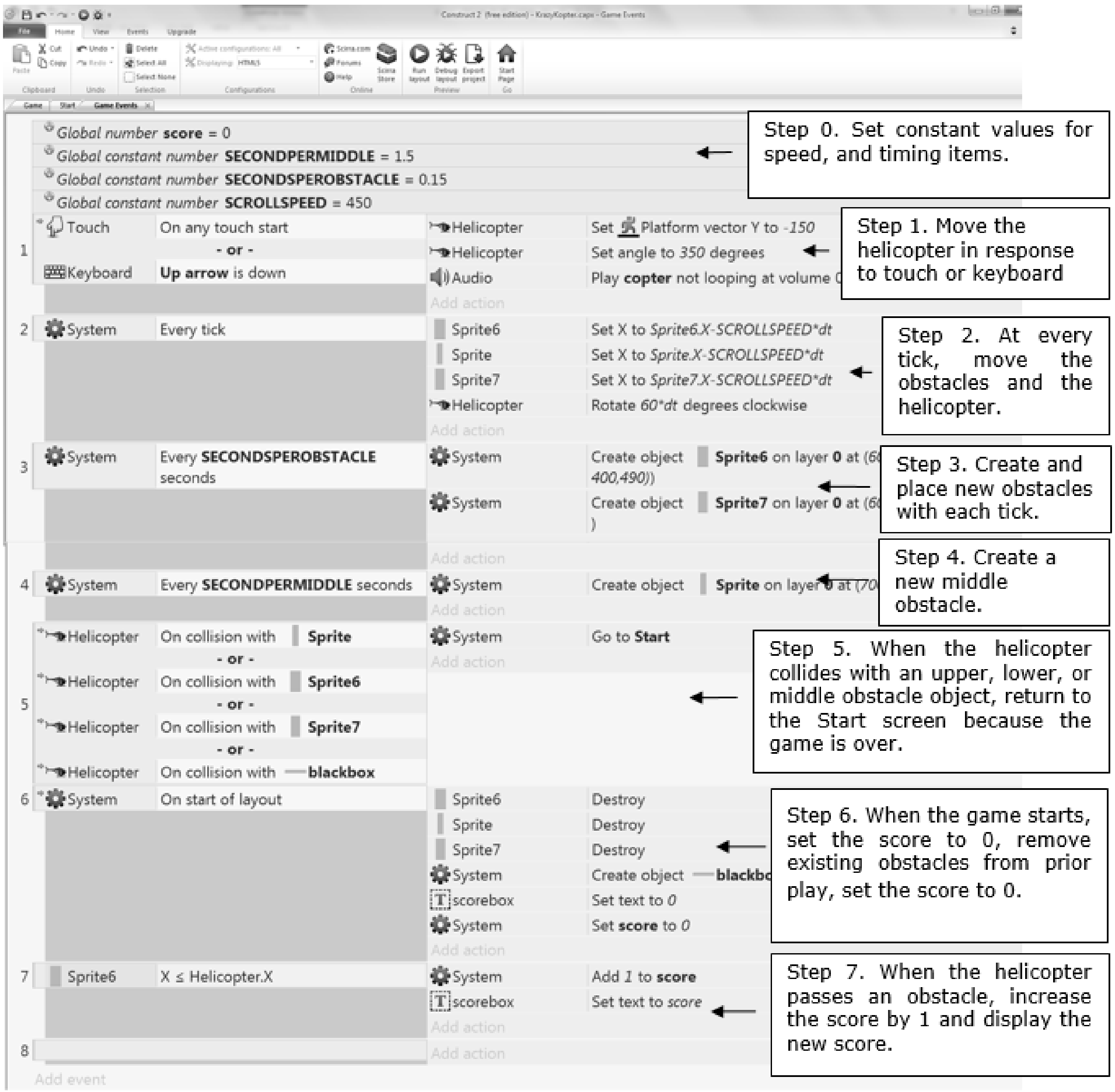
Editor's Note:

This paper was selected for inclusion in the journal as a EDSIGCon 2015 Meritorious Paper. The acceptance rate is typically 15% 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 2015.

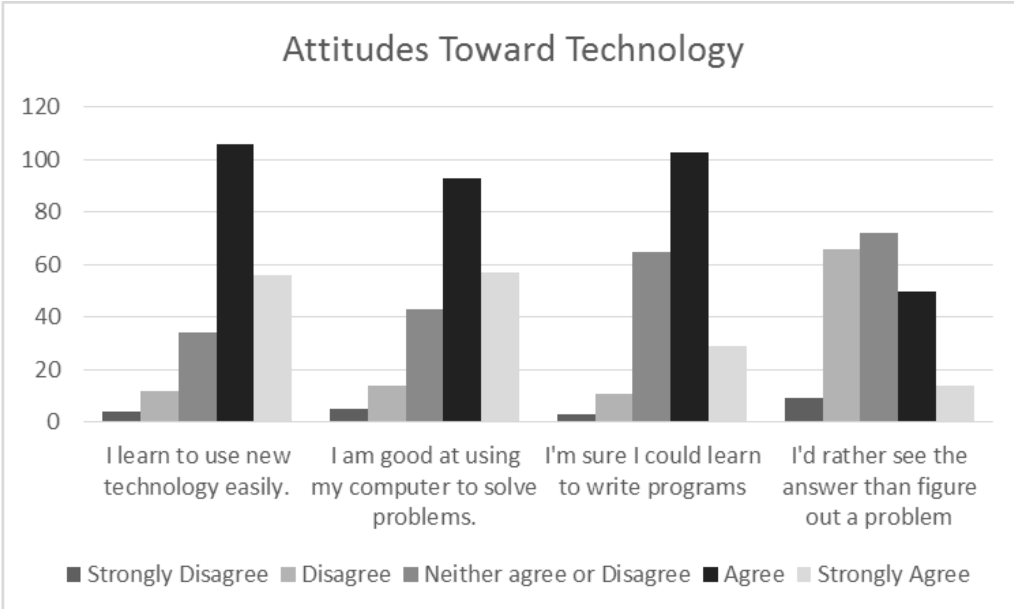
Appendix 1. Additional Figures



Appendix 1 Figure 1. (a) The start screen and game play screen when the game is playing on the web or a mobile device. (b) Designing the game play screen in Construct 2.



Appendix 1 Figure 2. Specifying rules of game play in Construct 2. Steps shown demonstrate programming concepts listed in Table 1.



Appendix 1 Figure 3. Attitudes toward technology among participants.

Appendix 1 Figure 4. ICT Literacy Experiences in Game Development.

Appendix 2. Gaming Resources

Table 1. Online Gaming Resources

Title	Web Address	Description
Construct 2	https://www.scirra.com/construct2	Creation engine for 2-D games that generates web and native applications.
Construct 2 Tutorials	https://www.scirra.com/tutorials/top	Tutorials for learning Construct 2.
Best Sound Effects Ever	http://www.bfxr.net/	Web app for creating and previewing sound effects for computer games. Sounds can be exported as WAV files.
Open Game Art	http://opengameart.org/	Open source collection of graphics for use as backgrounds, characters, shapes, and objects in computer games. Online forum for game developers to discuss and share their games and artwork.
Pixel Prospector	http://www.pixelprospector.com	Links to websites providing graphics for use in games without royalties, tools for creating sound and music, and additional resources for game developers.
Video Game Name Generator	http://videogamena.me/	Generates names for video games by randomly selecting unlikely word combinations.

For complete assignments, descriptions, and handouts for the gaming assignment, visit http://cis.bentley.edu/sandbox/ and search for Gaming Resources.

Appendix 3. Gaming Assignment

Working individually or with a partner, create your own 2-D Platform Game using Construct 2 sized to play on a mobile device. You might model it after the game we created in class, or look at one of the tutorials on the Construct 2 web site and model it based on one of those tutorials, or create something entirely original.

Basic Features

Your game should include several of these basic features:

- An attractive home screen with a button to play the game and rules describing how to play
- A game screen, that when the game is over, returns you back to the home screen
- Keyboard and touch input
- Spawned objects
- Music or sound
- Attractive graphics

Advanced Features

Your game should include several of these, or other advanced features:

- Allow user to turn on or off background music
- Score keeping
- Collision Detection
- Randomization of speed, placement of objects, etc.

Publish your Game to your Website

- Create a page called game.html and link to it from your website's home (index.html) page.
- Take a screenshot or two of your game, write a brief description of how to play, and describe which basic and advanced features above you included (and how).
- Include a link to a "full screen" version of your game.
- Place a link from your game.html page to a page named play.html that displays your game in full screen, ready to play.

Publish your Game to the Windows Store

Follow the instructions provided for how to publish your game to the Windows Store. This is an involved process, which you will need to complete in on a computer running Windows 8.

Grading:

Grading is based on CIS Sandbox tutor reviews and the instructor's comments. We will look for originality, playability, complexity, and aesthetics.

- **Acceptable:** Your game is very similar to the tutorial we did in class, but you changed the graphics or sounds. Not so original, but you get the idea.
- **Better:** Your game has a different play model, but it's not very playable. For example, you always win or lose immediately, the game play is not intuitive, or it's just not fun to play.
- **Best:** Your game is original, uses several basic and advanced features, and is highly addictive.

Salient Beliefs in Majoring in Management Information Systems: An Elicitation Study

Wallace Chipidza
wallace_chipidza@baylor.edu

Gina Green
gina_green@baylor.edu

Cindy Riemenschneider
c_riemenschneider@baylor.edu

Management Information Systems Department
Baylor University
Waco, Texas 76798, USA

Abstract

Research utilizing the Theory of Planned Behavior to understand behavior should first elicit beliefs about the phenomenon from the target population. In order to understand the reasons why students choose to major or not major in Management Information Systems (MIS), we elicited beliefs from 136 students attending university in the United States and in Zambia. We employed a questionnaire with open-ended questions to elicit beliefs about majoring in MIS. The gender split of study participants was 52%-48% with a female majority and their ages ranged from 19 to 35. Using content analysis of the generated qualitative data, we identified 11, 5 and 9 categories of behavioral, normative and control beliefs respectively. The results of our study indicate that student beliefs about the MIS major and profession have changed over the past decade; students now favorably perceive the MIS job market and attach importance to the opinions of industry professionals when making the decision to major in MIS. Analysis of the ranked elicited beliefs shows that most students believe that the MIS degree grants them competitive advantage in the employment marketplace.

Keywords: enrollment, management information systems, elicitation study, theory of planned behavior, information systems major, career

1. INTRODUCTION

Declines in enrollment into Management Information Systems (MIS) programs in the United States and elsewhere are well documented (Becerra-Fernandez, Elam & Clemmons 2010; Calitz, Greyling & Cullen, 2011; Huang, Greene & Day, 2008; Zhang, 2007). The implications of declining enrollments are many and varied: first, since Information Technology (IT) is a driving force for growth in advanced countries, if the decline continues, then the US risks falling behind other countries in technological development, to the detriment of the whole economy. Indeed, labor surveys show a dire shortage of skilled technology professionals worldwide (Manpower Group, 2012). Second, enrollment declines have a negative effect on business schools' tuition revenue, a situation that significantly impacts university operations since tuition is the primary revenue source for most universities (InsideHigherEd, 2013). Certainly, sharp declines in enrollment at some business schools have led to closure of their MIS departments (Aken & Michalisin, 2007). Last, declines in MIS enrollment affect graduates of MIS doctoral programs by shrinking their employment opportunities. Because declines in MIS enrollment have such dire consequences, it is important to understand the factors that influence a student's decision on whether to major in MIS.

Although various studies have been conducted to investigate the underlying causes of declining MIS enrollment, few studies have elicited input from prospective enrollees. By relying exclusively on influences garnered from the existing literature, studies that aim to understand why MIS enrollment rates are in decline may fail to capture newly emerging explanations for the phenomenon. In addition, certain influences that were major driving factors behind students' decisions in the past might have lost their potency with time. As such, a major thrust of our research study has been to learn from prospective MIS enrollees concerning the possible factors that lead them to make the decision whether to major in MIS. We employed the elicitation survey instrument that serves as input for the Theory of Planned Behavior (TPB) (Ajzen, 1985) in order to collect the most salient beliefs about MIS from students. This way, we can detect new beliefs that have gained salience in more recent years and might not be found in the existing literature. Moreover, utilizing an elicitation survey allows us to rank the beliefs and hence gain an understanding as to which beliefs are more instrumental in influencing the student's decision to major in MIS. Finally, our approach allows us to identify the most prominent social referents that influence the decision to major in MIS.

Our study is unique in that it is not limited to the United States; we also elicited beliefs about MIS from students in the Southern African country of Zambia. Since the Zambian government recognizes the role of IT in the transformation of its economy (ZICTA, 2009), the country will require educated MIS professionals to fuel the transformation process. Gathering beliefs about majoring in MIS from Zambian students should provide clues on which beliefs are likely to influence student enrollment in MIS in a developing country.

Our findings indicate that the most frequently cited beliefs about MIS are positive. However, and perhaps paradoxically, students report more barriers that make majoring in MIS a more difficult decision than the pull factors that make it an easier decision.

In the following sections, we present the theoretical foundation for our study followed by a discussion of the method of data collection, analysis, and results of the analysis. We conclude by discussing the research contributions and limitations.

2. THEORETICAL FOUNDATION

An extension to the Theory of Reasoned Action (TRA) (Fishbein & Ajzen, 1975), the TPB aims to explain the link between the intention to perform a stated action and the actual behavior that proceeds from the intention. Like the TRA, the TPB asserts that an individual's attitude towards a behavior and perceived subjective norms (external pressures to perform a behavior) both influence the intention to perform a certain behavior. Unlike the TRA however, TPB contains an additional construct, the perceived behavioral control (PBC) of the individual. PBC is the individual's belief that she is able to perform a certain behavior. PBC is important because certain behaviors are not under an individual's complete volitional control; such behaviors might require additional resources that are beyond the reach of the individual.

Students' intentions to enroll in MIS have been investigated under the lens of the TRA, most notably in a study by Zhang (2007). As Zhang (2007) explains, a student with a favorable attitude towards MIS is more likely to enroll in MIS than one who does not have a favorable attitude. Moreover, if a student's social influences are supportive of the student's decision to enroll in MIS, then the likelihood that the student will do so increases. However, Zhang (2007) discounts the possibility that students might not have full control of the decision to enroll, as he asserts that the decision to major in MIS is one that students can make of their own volition. That assertion might not hold under all circumstances, however, because a student might want to enroll in MIS and might not be able to afford it. In developing countries such as Zambia, lack of access to computers might prevent a student from majoring in MIS. Furthermore, a student might feel that the MIS major is difficult relative to other majors. Indeed, previous studies have shown that the perceived difficulty of a major is a barrier to its enrollment (Noble Calkins & Welki, 2006; Saemann & Crooker, 1999) and that this is especially true of MIS (Locher, 2007). These barriers indicate that enrolling in MIS is a decision that might not be completely under the control of a student. Therefore, we feel that PBC is a valid construct that can explain additional variance for the intention to enroll in MIS.

To examine behavior using the TPB an elicitation survey completed by a subset of participants prior to creating the final survey instrument is recommended (Ajzen, Nichols & Driver, 1995; Sutton, French, Hennings, Mitchell, Wareham, Griffin, Hardeman, & Kinmonth, 2003). However despite the prevalence of studies that utilize the TPB/TRA theoretical framework, few of them actually conduct an elicitation survey (Sutton et al., 2003). This elicitation stage is important because it allows for the identification of cognitive and affective salient beliefs resident in the phenomenon under investigation, hence forming a foundational stage of TPB research (Ajzen et al., 1995). Specifically, respondents are surveyed on three aspects of their beliefs: behavioral, normative, and control beliefs. Behavioral beliefs indicate whether an individual holds a favorable view of the behavior under investigation. Normative beliefs measure the social pressure that an individual feels to perform that behavior. Control beliefs assess how much control the individual feels she has over the decision to perform the stated behavior. Therefore, in the context of understanding students' decision to major in MIS, our study aims to uncover students' attitudes towards the MIS major, their perceptions of sources of social pressure and the amount of control they feel they have over the decision to major or not to major in MIS.

3. METHOD

We elicited beliefs from the general population of students enrolled in business classes at two private universities in the US and Zambia. We solicited responses by enlisting the help of professors who offered extra credit to their students as incentive for completing the survey by a specified date. The approximately 90% response rate was satisfactory and as such, we did not send reminders to potential respondents. Participants entered their student IDs on the survey. These IDs were sent to professors to assign extra credit points; however the IDs were not used in our analyses nor were responses sent to professors, all in order to maintain anonymity. Our questionnaire employed questions formulated by Ajzen and Fishbein (1980) and Ajzen and Driver (1991), but modified to suit the context of our study. The questions evaluated students' behavioral, normative and control beliefs concerning the decision to major in MIS. For the behavioral beliefs, we did not restrict our study to cognitive beliefs, but we also elicited affective beliefs from our respondents. For example, for eliciting the positive cognitive beliefs about majoring in MIS we asked: "*What are the advantages of majoring in MIS?*" For the positive affective question we asked: "*What do you like about majoring in MIS?*" We include the complete set of elicitation survey questions in Appendix A.

Responses were elicited in an open-ended format that allowed respondents to freely articulate their beliefs concerning majoring in MIS. For each question we included five response lines for respondents to fill in as prior research has shown that few respondents supply more than five beliefs (Sutton et al., 2003).

For all questions, an initial round of coding was undertaken on a subset of the data using content analysis to uncover themes within the data, resulting in categories into which responses fell. We proceeded to code the full response set based on the identified categories. Two researchers independently coded the full set of response data; the inter-rater reliability was 90%.

4. RESULTS

Out of 136 returned surveys, 110 were usable. There was a gender split of 52%-48% with a female majority with ages ranging from 19 to 35. Up to 75% of the respondents identified their socio-economic status as upper middle class, 16% as upper class and 9% as lower middle class. The college level breakdown was as follows: 34% freshman, 23% sophomore, 30% juniors and 13% seniors. We summarize the demographic information of the respondents in Table 1 of Appendix B.

We show the descriptive statistics of the results in Appendix B, Table 2. Cohen's kappa (Cohen, 1960), which measures the rate of agreement whilst discounting the agreement due to chance between independent raters of categorical data in qualitative analysis, varied from 61% to 100% for the different questions. Using the Fleiss (1981) benchmarks, inter-rater agreement was good in

43% of the categories and excellent in the remaining 57% of categories.

We carried out an independent samples t-test in SPSS to find out if there were any gender differences in the number of responses supplied per individual. We found no significant gender differences except in the category of advantages of majoring in MIS; on average, female respondents supplied more advantages than male respondents; the difference was significant at the 10% level.

The following subsections describe the results of our analysis in detail.

Behavioral beliefs
Many of our elicited behavioral beliefs are consistent with previous IS career choice studies. Students perceive that MIS bestows competitive advantage; 84% of respondents mentioned competitive advantage as a motivator for enrolling in MIS. These students recognize that attaining a job is a competitive endeavor that requires them to acquire assets that set them apart from other job seekers. Specifically, students commented that majoring in MIS "*looks good*" on a resume and "*(helps one) stand out from other graduates*." The competitive advantage perceived by students was not limited to just the job application phase. According to one participant, enrolling in MIS would give an employee "*an advantage against other employees*." This suggests that MIS is perceived to be beneficial even beyond the job attainment milestone as it grants employees increased chances for upward mobility in the future. However, our elicited beliefs indicate that perceived competitive advantage may not extend to under-developed countries as exemplified in a comment from a Zambian study participant bemoaning the job situation in her country: "*There are few jobs for IT in Zambia*."

Furthermore, MIS is perceived as relevant for today's world. Over 80% of respondents identified technical skill acquisition as an advantage accrued from majoring in MIS. Some of the responses that fell in this category emphasized the increasingly vital role of information technology in advancing businesses in different industries. As one respondent stated: "*(an advantage of enrolling in MIS is the) ability to use information received about operations to improve the company*." Technology use was also a prominent theme in this category. Multiple responses were exemplified by this one quote regarding an advantage of majoring in MIS: "*Ability to not worry whether I would be able to use technology in the work place*." Students therefore associated majoring in MIS with gaining a skill that is not only useful for performing work tasks, but one that also alleviates anxiety stemming from using unfamiliar technology.

Our study revealed that MIS is perceived to be a high salary industry that affords employees sufficient job security. Forty percent of respondents described the salary earned by MIS professionals as "*good*", "*high*" or "*competitive*" and therefore an advantage of majoring in MIS. Personally-rewarding factors, such as the perception that in MIS, "*(there are) always new things to learn*" and the view of the MIS profession as "*fun (and) entertaining*", also emerged as encouraging factors for adopting the MIS major; these personally-rewarding factors were mentioned by 34% of respondents. In addition, work environment factors, mentioned by 12% of respondents, such as "*(the ability to) work from home*" and "*(working with) smart people*" emerged as positive motivators for studying MIS.

On the other hand, some elicited beliefs suggest MIS suffers from an image perception problem. Three in every four respondents mentioned common negative perceptions of the major that might discourage students from enrolling. Respondents spoke of the perceived "*nerd stereotype*" and "*little human interaction*" associated with MIS professionals, or that the major is boring and "*It's hard …to get excited about*." In addition, respondents characterized the work environment for MIS professionals as "*high stress*" and "*too sedentary*." These perceptions of MIS are likely to strongly discourage enrollment.

Moreover, there exists a pervasive perception that MIS is difficult. Multiple responses mentioned by 75% of respondents articulated how challenging the major is. Many answers in this category succinctly described the MIS major with the synonyms "*hard*", "*tough*" and "*challenging*." A more descriptive response was the following: "*(MIS has) core classes that might be beyond my capabilities*." Other responses hinted at what is difficult about MIS: "*MIS seems like sometimes it would be difficult to understand because it deals with so much technology*." This fits with another category of disadvantages of majoring in MIS; almost 30% of respondents identified "*programming*" or "*coding*" as a reason for not enrolling in MIS because "*learning to program can be difficult*." Indeed, it seems that MIS and programming are indistinguishable in the minds of many students.

Majoring in MIS is perceived as time consuming. Perceptions such as "*too many classes*" required for graduation and classes that entail "*spending long hours on assignments*" were prevalent. Forty-six percent of study participants felt that enrolling in MIS required more classes than the average as exemplified by this response: *"(MIS requires) more classes to take so it would be more expensive (than other majors)*." Furthermore, enrolling in MIS is perceived to limit time available for other activities because students perceive that the major requires "*lots of time outside of class working on the material*." There was also a perception articulated by 44% of respondents that the cost of learning technology would persist beyond school since keeping up with new technology developments requires significant ongoing effort. As two respondents stated, "*(in MIS) knowledge becomes outdated easily*" and "*IT is evolving, so the information may not be relevant in the future*." Students might potentially feel that investing many resources into a rapidly changing field is not worthwhile.

Lack of interest also discourages students from majoring in MIS. Over 40% of the respondents mentioned that they simply were not interested in the MIS field or that they had passion for other majors. Sample responses in this category included "*I am not very interested in technology*", "*I don't enjoy the subject*" and "*(I) couldn't major in finance (if I majored in MIS)*." These responses suggest that for many students, MIS is a boring major that inspires little excitement and is thus not feasible as a profession. Table 3 of Appendix B summarizes the most salient behavioral beliefs about MIS.

Subjective norms

Subjective norms refer to sources of societal influence that are instrumental in the decision-making process of an individual, given a set of choices. Five broad groups of social referents emerged as influential in the decision to enroll in MIS: career counselors, educators, MIS professionals, friends, and family. Just over 70% of participants indicated that the opinions of family members are instrumental in the decision to major in MIS. Sample answers in the family category included "*parents*", "*sister*" and "*grandpa(rent).*" For the friends' category, responses ranged from "*friends*" and "*boyfriend*" to "*fraternity friends*" with explicit names of fraternities and sororities suggesting that the decision to major in MIS can be influenced even in the college years. Our study elicited a referent not often cited in previous studies: MIS professionals. Over 20% of participants indicated that exposure to MIS professionals have an impact on their decision to major in IS. The opinions of high school career counselors also influence the decision to major in MIS, as shown by the 15% of respondents who mentioned them as instrumental in the decision making process. We present the summary statistics for subjective norms in Table 4, Appendix B.

Perceived Behavioral Controls

Several responses to the control questions echoed the answers to the belief questions. Respondents stated that "*the average starting salary of MIS majors*" and "*more job opportunities*" in MIS make it easier for an individual to enroll in MIS.

Conversely, the perceived time-consuming nature of the MIS major appears to be a personal cost that makes the decision to major in MIS harder. Unsurprisingly, therefore, 46% of respondents stressed that if MIS required less additional effort than other majors then perhaps students would be more willing to major. However, additional insight not emphasized in previous research was gathered from responses to the control questions. For example, 12% of respondents mentioned the presence of a support system as a pull factor to majoring in MIS. As one respondent stated "*my parents would support me (in this decision)*." Also falling in the support system category was availability of tutors and tutorials, convenient (or lack of) access to computers, and the availability of "*good MIS professors*" who would make the decision to major in MIS easier. The latter response echoes the family and educators categories of the previous section and therefore represents an overlap between the control and normative beliefs. Seven percent of respondents mentioned that MIS affords students valuable experiences such as *"getting to learn new things."* Finally, majoring in MIS seems to grant students the ability to distinguish themselves from others as evidenced by this response: "*(MIS allows me to) distinguish myself from other students.*" These factors make the decision to major in MIS easier. On the other hand, low self-efficacy in the form of students feeling that they did not have the intellectual ability to succeed in the MIS major, was mentioned by 38% of participants as a significant deterrent to majoring in MIS. Moreover, Zambian participants mentioned the lack of access to computers and reliable Internet as significant deterrents to majoring in MIS. The summary of results is shown in Table 5, Appendix B.

5. DISCUSSION: CONTRIBUTIONS AND IMPLICATIONS

This study contributes to our understanding of why students choose to major or not major in MIS. First, our study reveals several new beliefs found by eliciting beliefs from students rather than drawing from the currently documented beliefs in extant literature. Prior studies identified concerns over job availability for MIS graduates as a major contributor to declining enrollment in MIS (Foster, 2005; Lomerson & Pollacia, 2006; Mahmoud, 2005). Furthermore, a widespread perception that IT jobs are moving offshore also discouraged students from majoring in MIS (Foster 2005; Locher, 2007). However, our study shows that students currently hold a more favorable perception of the MIS job market; MIS is now viewed as a major that affords its graduates many job opportunities. In fact, 84% of respondents mentioned that the MIS major grants its graduates sufficient job security and competitive advantage in the marketplace. Clearly, students' perception of MIS jobs' availability has sharply changed over the past decade.

Second, our study reveals new beliefs concerning external influences on the decision to major in MIS. Past studies have identified the opinions of family, educators (Zhang, 2007), friends (Downey, McGaughey & Roach, 2009) and career counselors (Noble Calkins & Welki, 2006) as contributory factors in the decision to major in MIS. Our findings largely confirm these influences to be valid, but we also found an influential group that was largely unexplored in the literature. More than 20% of respondents mentioned that the opinions of MIS professionals also influence the decision to enroll in MIS; this group actually emerged as the third most influential, surpassing even the influence of educators and career counselors. The belief that the opinions of MIS professionals matter in the decision to major in MIS might reflect the fact that technology professions are gaining prominence; students might actually personally know more MIS professionals than was the case in the past. As a result, inviting MIS professionals to speak to students in high school in class or at career fairs might create a favorable impression of the MIS profession and hence improve student enrollment into the major.

Third, eliciting beliefs regarding control in the decision to major in IS found that access to support systems is an increasingly important factor in deciding to major in MIS. Thus departments should be encouraged to develop a consistent supply of tutors and offer access to tutorials to ensure potential majors feel that ample resources are available for success.

As a fourth contribution, we have presented a ranking of the most salient beliefs about MIS that students possess. Previous studies identified various factors that influence enrollment into MIS programs, but few of them explore the relative importance of each factor. If we take the percentage of respondents that expressed a particular belief as a proxy for salience of that belief, then our study shows that certain beliefs are more salient than others. By this measure, the most alluring reason for enrolling in MIS is the competitive advantage that the major grants its graduates in the employment marketplace. The second most-cited advantage was the acquisition of technology skills. This finding is important in that it supports the encouragement of MIS as a complementary second major for many business students. On the other hand, the image of MIS as a profession that involves little to no human interaction emerged as the single most salient factor that discourages enrollment into MIS; hence underscoring the caricature of the MIS profession as entailing little more than sitting behind a desk and staring at a computer screen for hours on end. Closely trailing the human interaction factor is the perceived difficulty of the MIS major, which has already been identified in prior studies as a significant deterrent to majoring in MIS (Locher, 2007). Required preparatory classes in MIS at the high school level could help remedy the perception that the MIS major is difficult.

Given the many favorable views expressed by students about the MIS major, the question then arises why MIS enrollment rates keep declining. Since our study included students from a variety of majors, we can conclude that positive views about MIS are not limited to MIS students, but persist throughout the population of college students. However, since these views are not translating into increased enrollment in MIS programs, it is possible that students only become aware of the MIS major and its positive characteristics after starting college. At that stage, students might already have made the decision to enroll in other majors, suggesting that the high school preparation phase is influential in the decision to major in MIS. Another explanation is that the unfavorable characteristics of the major simply outweigh its favorable attributes in importance to the students, resulting in students adopting other majors. As has been shown in various studies, the existence of a negative belief towards a particular behavior may prevent its subsequent performance (Darker, Larkin, & French, 2007; Hancock, 2013). For example, a student might be attracted to MIS because of the high starting salary but at the same time be repelled by its perceived difficulty. Therefore MIS educators and career counselors should focus not only on highlighting the positive aspects of majoring in MIS, but also on transforming the negative perceptions that plague the major to positive perceptions (Sutton, 2002). If the major is perceived to be difficult, for example, educators and counselors may explain to students that difficult majors are more prestigious, have less competition for jobs and therefore award higher salaries than easier ones. Moreover, since students explained that the difficulty in MIS stems from its programming component, educators and counselors may clarify that MIS encompasses much more than just programming, and that MIS professionals are not limited to just writing code. In this manner, a previously perceived disadvantage could now be seen as an advantage.

The rapidly changing nature of MIS relative to other fields imposes a cost that discourages enrollment into the major. An often-mentioned observation by our study participants is that new programming languages emerge frequently and MIS professionals therefore need to be in a constant state of learning, a requirement that is limited to only a few majors. This stands at odds with other more traditional majors, where students go to college, graduate with a bachelor's degree and at that stage they would have fulfilled the bulk of their learning requirements. With the cost of learning persisting beyond school, it is little surprise that many students who might otherwise be willing to major in MIS would be deterred from doing so. However students can be informed that many top-tier professions such as accounting, law, medicine, etc. require ongoing learning of their professionals.

Finally, our analysis suggests differences in beliefs between study participants in the US and Zambia. Whereas American participants highlighted job availability as an advantage of majoring in MIS, some Zambians expressed concern that the availability of IT jobs is limited. There seems to be a gap between the Zambian government's expressed vision for the role of IT in transforming its economy and students' perception of that vision. Furthermore, some Zambian respondents mentioned the lack of computers and Internet access as an inhibitor for MIS education, a concern that was not articulated by even a single American participant. A Zambian majoring in MIS therefore incurs higher cost than her American counterpart since it's more difficult for her to obtain a computer and Internet access. Consequently, Zambians may feel less control over the decision to enroll in MIS than their American counterparts. In order to encourage student enrollment in MIS in Zambia and similar countries, reliable computing and Internet infrastructure will have to be available in schools.

6. CONCLUSION: LIMITATIONS AND FUTURE WORK

Our study has limitations: our sample of respondents is wealthier than the general population, presumably because our subjects all attend private universities in both Zambia and the US. Moreover, 43% of respondents were at the junior or senior levels of college. Wealthy and educated respondents could be more aware of MIS. To address these limitations, future work could survey a sample that is more representative of the general population by including students from public universities. Furthermore, our Zambian sample was disproportionately smaller than its American-based counterpart; this limits our ability to make generalizable comparisons between students based in these two nations. Nonetheless, we found it useful to highlight responses from the Zambian respondents so that we can acquire an understanding of which factors influencing the decision to major in MIS may differ between the two countries. Finally, future work could fulfill the purpose of eliciting salient beliefs and use the results of the elicitation stage to conduct a full study that investigates enrollment in MIS under the TPB framework.

7. REFERENCES

Ajzen, I., & Fishbein, M. (1980). Understanding attitudes and predicting social behavior. Englewood Cliffs, NJ: Prentice-Hall.

Ajzen, I. (1985). From intentions to actions: A theory of planned behavior. In J. Kuhl & J. Beckmann (Eds.), *Action control: From cognition to behavior*. Berlin, Heidelber, New York: Springer-Verlag. (pp. 11-39)

Ajzen, I., & Driver, B. L. (1991). Prediction of leisure participation from behavioral, normative, and control beliefs: An application of the theory of planned behavior. *Leisure Sciences, 13*(3), 185-204.

Ajzen, I., Nichols, A. J., & Driver, B. L. (1995). Identifying Salient Beliefs About Leisure Activities: Frequency of Elicitation Versus Response Latency1. *Journal of Applied Social Psychology*, *25*(16), 1391-1410.

Aken, A., & Michalisin, M. D. (2007, April). The Impact of the Skills Gap on the Recruitment of MIS Graduates. In *Proceedings of the 2007 ACM SIGMIS CPR Conference on Computer Personnel Research: The Global Information Technology Workforce* (pp. 105-111). ACM.

Becerra-Fernandez, I., Elam, J., & Clemmons, S. (2010). Reversing the Landslide in Computer-Related Degree Programs. *Communications of the ACM*, *53*(2), 127-133.

Calitz, A. P., Greyling, J. H., & Cullen, M. D. (2011, October). The Accreditation of ICT Degree Programs in South Africa. In *Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment* (pp. 49-58). ACM.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: the Kappa Statistic. *Computational linguistics*, *22*(2), 249-254.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37-46.

Darker, C. D., Larkin, M., & French, D. P. (2007). An Exploration of Walking Behaviour — an Interpretative Phenomenological Approach. *Social Science & Medicine*, *65*(10), 2172-2183.

Downey, J. P., McGaughey, R., & Roach, D. (2009). MIS versus Computer Science: An Empirical Comparison of the Influences on the Students' Choice of Major. *Journal of Information Systems Education*, *20*(3), 357.

Fishbein, M., & Ajzen, I. (1975). Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research. Reading, MA: Addison-Wesley.

Fleiss, J. L. (1981). The Measurement of Interrater Agreement. Statistical methods for rates and proportions, 2, 212-236.

Foster, A. L. Student Interest in Computer Science Plummets: Technology Companies Struggle to Fill Vacant Positions. (2005). *The Chronicle of Higher Education: Information Technology*, *51*(38), A31.

Hancock, J. A. (2013). *Exploration of Five Condom-related Behaviours in the UK: Development and Evaluation of Theory-based Online Safer Sex Intervention* (Doctoral dissertation, Coventry University).

Huang, W. W., Greene, J., & Day, J. (2008). Outsourcing and the Decrease of IS program Enrollment. *Communications of the ACM*, *51*(6), 101-104.

Locher, M. (2007). IT Education: Where Have All the Young Geeks Gone. *CIO*, *20(15)*, 49-53.

Lomerson, W. L., & Pollacia, L. (2006). Declining CIS Enrollment: An Examination of Pre-College Factors. *ISEDJ*, *4*(35).

Mahmoud, Q. H. (2005). Revitalizing Computing Science Education. *Computer*, (5), 100-98.

Manpower Group. 2012 Talent Shortage Survey. (2012). Retrieved June 7, 2015 from http://www.manpowergroup.us

Martin, C. D. (1998). Is Computer Science a Profession?. *ACM SIGCSE Bulletin*, *30*(2), 7-8.

Moody's Warns of Impact of Enrollment Declines | InsideHigherEd. (2013, September 10). Retrieved July 1, 2015 from https://www.insidehighered.com

Noble Calkins, L., & Welki, A. (2006). Factors that influence choice of major: why some students never consider economics. *International Journal of Social Economics*, *33*(8), 547-564.

Saemann, G. P., & Crooker, K. J. (2000). Student Perception of the Profession and Its Effect on Decisions to Major in Accounting. *Journal of Accounting Education*, *17*(1), 1-22.

Sutton, S. (2002). Using Social Cognition Models to Develop Health Behaviour Interventions: Problems and Assumptions. in Rutter, D. & Quine, L (Eds.), *Changing Health Behaviour: Intervention and Research with Social Cognition Models*. Buckingham: Open University Press. 193-208.

Sutton, S., French, D. P., Hennings, S. J., Mitchell, J., Wareham, N. J., Griffin, S., Hardeman, W., & Kinmonth, A. L. (2003). Eliciting Salient Beliefs in Research on the Theory of Planned Behaviour: The Effect of Question Wording. *Current Psychology: Developmental, Learning, Personality, Social, 22,* 234-251.

Zambia Information and Communication Technology Authority (ZICTA) Site. (2009). Retrieved July 1, 2015 from http://www.zicta.zm

Zhang, W. (2007). Why IS: Understanding Undergraduate Students' Intentions to Choose an Information Systems Major. *Journal of Information Systems Education*, *18*(4), 447.

Appendices and Annexures

A. Survey Questions: Elicitation Survey for Identifying Reasons Why Students Choose to Major in Management Information Systems

1. a) What is your age?
   b) What is your gender?
   c) What is your college classification? (Freshman/First Year, Sophomore/Second Year, Junior/Third Year or Senior/Fourth Year)
   d) What's your city and state of origin?
   e) In what socio-economic group would you describe yourself? (Lower, Lower- Middle, Upper-Middle, Upper)
2. What do you believe would be the *advantages* of you majoring or double-majoring in MIS?
3. What do you believe would be the *disadvantages* of you majoring or double-majoring in MIS?
4. What would you *like or enjoy* about majoring or double-majoring in MIS?
5. What would you *dislike or not enjoy* about majoring or double-majoring in MIS?
6. What *thoughts* come to your mind about a major or double-major in MIS? (book)
7. Are there any individuals or groups who would think that you *should* major or double-major in MIS?
8. Are there any individuals or groups who would think that you *should not* major or double-major in MIS?
9. If any such individuals or groups come to mind if you considered majoring or not majoring in MIS, please *list them* below. (book)
10. What would make it *difficult* for you to major or double-major in MIS?
11. What would make it *easy* for you to major or double-major in MIS?

B. Tables

	Category	Frequency	Percent of Sample
Gender	Female	57	52%
	Male	53	48%
College Level	Freshman	37	34%
	Sophomore	25	23%
	Junior	33	30%
	Senior	15	13%
Socioeconomic Status	Lower Middle	10	9%
	Upper Middle	82	75%
	Upper	18	16%
University	American	106	97%
	Zambian	3	3%

Table 6: Demographic Information

Beliefs	Question	Total Beliefs	Mean (SD) beliefs per person	Percent of people who gave 3 or more beliefs	Cohen's Kappa
Behavioral (attitude towards major)	Like or enjoy	263	2.39 (1.64)	39	0.86
	Advantages	384	3.49 (1.41)	77	0.70
	Dislike or hate	206	1.87 (1.72)	29	0.61
	Disadvantages	252	2.29 (1.78)	39	0.75
Control (barriers, facilitators)	Easy	180	1.64 (1.39)	20	0.81
	Difficult	216	1.96 (1.46)	13	0.85
Normative Referents	Individuals, Groups	146	1.42(1.67)	21	1.00

Table 2. Descriptive Statistics for Elicited Beliefs

Category	Number of Respondents	Percent of Respondents
Competitive Advantage	92	84%
Acquisition of Technical Skills	90	82%
Negative Image	83	75%
Difficult classes	82	75%
Time-consuming	51	46%
Constantly-changing technology	48	44%
Lack of interest	47	43%
High salary	43	39%
Personally Rewarding	37	34%
Difficulty with programming	31	28%
Positive work environment	13	12%

Table 3: Behavioral Beliefs

Category	Number of Respondents	Percent of Respondents
Family	24	71%
Friends	14	41%
MIS/IT Professionals	7	21%
Advisors	5	15%
Educators	2	6%

Table 4: Normative Beliefs

Category	Number of Respondents	Percent of Respondents
Time-consuming	66	64%
Too much additional effort	47	46%
Lack of interest	39	38%
Lack of ability/self efficacy	39	38%
Job opportunities, placement	20	21%
Individual Support system	12	12%
Valuable experience	7	7%
Distinguish from others	6	6%
High salary	4	4%

Table 5: Control Beliefs

Streamlining the Capstone Process: A Time-Saving Approval System For Graduate Theses/Projects

James Grooms
jag03rd@gmail.com
Engineering Software Solutions
Wilmington NC 28401

Douglas Kline
klined@uncw.edu

Jeffrey Cummings
cummingsj@uncw.edu

Information Systems / Operations Management
University of North Carolina Wilmington
Wilmington NC 28403

Abstract

Capstones have become an integral part of many information systems programs, both at the undergraduate and graduate level. One of the challenges can be tracking the process from the start of the capstone to completion. This paper describes the analysis, design and implementation of a web application for the approval workflow of a master's program in information systems. The system replaces a paper form based process that was confusing, time-consuming, and error-prone. The system uses asynchronous JavaScript, responsive design, and clickable email links to provide a native-like look-and-feel on mobile devices, and reduce approval time. Student statuses are stored in a relational database, and program-level reports are provided for administrative decision making.

Keywords: capstone management, software development, web application, mobile development, responsive design, asynchronous JavaScript, single page application.

1. INTRODUCTION

Program capstones are common occurrences in various levels of Information Systems degrees whether that be creating a capstone course for undergraduates (Schwieger & Surendran, 2011) or completion of a thesis/capstone for graduate programs (Kline, et al, 2012). They provide students the opportunity to expand on what they have learned in the program and apply this knowledge to a real world project (Bruhn, 2004) while often times interacting with external clients (Reinicke & Janicki, 2011). As rewarding and important as this requirement is, administration and completion of these projects can be problematic, often without a clear process (Goold, 2003, Novitzki, 2001). Furthermore, faculty supervising the capstone project often take various approaches to supervision which can cause challenges in tracking where the capstone is in the university process (Perez, et al. 2012). While there have been some tools suggested for managing the actual project being undertaken (Olarte, et al. 2014), there still remains issues surrounding the administration of the capstone process itself. Issues include various signatures on multiple forms from faculty and directors as well as the ability to track students throughout the process.

This paper presents a system for managing the workflow of thesis/capstone project requirements for a graduate program in information systems. The system was completed as a capstone project degree requirement for the program in the spring of 2014. The project took approximately one year and incorporated technologies such as AJAX and Bootstrap, resulting in a web application that has a native look-and-feel on a wide range of devices. The main goals were to help guide students through the process, and ease the burden of obtaining approvals for the steps in the process.

For the current master's program, students have to seek approvals throughout the various stages of the thesis/capstone project. These include:

- A faculty member's agreement to chair the committee
- Multiple faculty members' agreement to be on the committee
- Committee members' agreement to a proposal date
- Committee members' agreement to the defense date

These approvals are typically achieved through hand-written signature on a physical form. Since students only perform this once, they are unfamiliar with the process, complicating matters. The physical forms are recorded in a spreadsheet and filed in physical file folders, one for each student. This made it difficult to answer questions such as:

- Which students have completed their proposal, but not their defense? (for the director)
- How many committees am I on? (for a faculty member)
- Where is my approval form? (Who am I waiting on?) (for a student)

The remainder of the paper describes the requirements analysis, design, and implementation of a system to streamline the approval process, and provide decision making information to students, faculty and staff. Section 2 describes the current manual process, and lays out the requirements of the new system. Section 3 presents the architecture of the completed system, and the technical design decisions made along the way. Section 4 reviews the completed system and its implementation process, as well as the resulting benefits.

2. ANALYSIS

Current Process

The legacy process involved the following forms, in number sequence, with required signatures:

1. Capstone/Thesis registration form
   - Student
   - Committee Chair
   - Director
2. Establish Committee form
   - Committee Chair
   - Committee Members (2 +)
   - Director
3. Proposal Scheduling form
   - Student
   - Chair & Committee Members
   - Director
4. Proposal Approval
   - Chair & Committee Members
   - Director
5. Defense Scheduling form
   - Student
   - Chair & Committee Members
   - Director
6. Final Defense Approval
   - Chair & Committee Members
   - Director

Four of the six forms (1, 2, 3 & 5 above) are for informational use only, creating some formality and ensuring awareness of events by all stakeholders. Without the forms, miscommunication amongst the stakeholders on dates and committee membership was common. Signatures for these four forms were the responsibility of the student, entailing finding each person individually. In recent years, the forms had been created as digitally-signable Portable Document Format (pdf) files, with the intention that the file be email-routed. However, not everyone could digitally sign the document, and it would get lost in email inboxes. As a result, the pdf files were not consistently used.

The most important forms in the process are the Proposal Approval (#4 from list) and Final Defense Approval (#6 from list). Signatures on these forms indicate completion of a degree requirement. These two forms were typically signed immediately following a proposal or defense, when all committee members were together in a single room.

After all signatures were obtained, the program coordinator entered individual form information into a common spreadsheet. The paper forms ended their trip in physical file folders, one for each student. The spreadsheet was used for quick one-student status lookups such as "Has Jane Smith proposed? Who is her chair?"

The legacy manual process had many problems. First and foremost was trying to achieve compliance from the students and faculty. Even though the processes were well documented and in a Student Handbook, many students and faculty were recurrently unaware of the forms and processes to complete the capstone. Faculty would forget whose committees they were on. Students had difficulty obtaining signatures promptly. Forms would remain on a faculty member's desk for weeks, or get lost entirely. Forms were commonly filled out after-the-fact, which is contradictory toward the purpose of the forms -- to notify of future events. Notification of future events also affected the degree requirement that proposals and defenses be public. We had to ask: If there is no future notification of a defense, can we still consider it to be public?

Another problem of the legacy process was a lack of reporting capability. There was no easy way to aggregate across students for reporting and decision making. For example, "how many students have proposed but not defended?" would require a manual count in the spreadsheet (if it was up-to-date) or pulling all students files individually.

In summary, the existing process was chaotic and frustrating for all stakeholders. Thus, the decision was made to eliminate the manual process and design/build a new system for tracking and approving capstones/theses.

Design Goals

The new system was meant to benefit not only faculty and students but those responsible for tracking and coordinating the capstone/thesis process (e.g. program administrators). Under the current approach, students coordinate with the faculty and program administrators to complete their capstone project or thesis. Signatures on the paper forms document that faculty were in agreement on committee membership as well as event dates and times. However, students bore the brunt of reaching consensus among parties and gathering signatures. Because of graduation deadlines, this could potentially have serious consequences.

Because of the chaotic and inconsistent nature of the legacy process, it was not possible to establish metrics or quantify the process. Thus, we were unable to establish quantifiable goals such as "reduce the in-process time for form B by 20%". We anticipated that, based on the stakeholders' feedback, that any improvement would be welcome.

The first design goal for the system was to reduce in-process time for an approval. Students still had to reach consensus among party members, but there shouldn't be delays due to the paperwork and getting signatures. Reducing in-process time would save time for all stakeholders.

The second goal was to remove the need for physical signatures. For most of the approvals, a "legally-binding" physical signature was unnecessary; evidence of notification and approval would be sufficient. Forms 1-3 and 5 above are all examples of unnecessary physical signatures. If we could document notification and approval, then physical signatures would be unnecessary.

The third design goal was to provide some status reporting for all the stakeholders. Reports for all stakeholders needed to be based on the same information. Students needed to know their current status, and their position in the process. Faculty needed to know their committee obligations and upcoming proposals and defenses. Program administrators needed program-level aggregate reports to make decisions.

Architecture Decisions

These design goals led to several key decisions regarding the system. First, the system would be "in the cloud", in the form of a database-driven web site. A common database would keep all stakeholders up-to-date with the same information. Second, emails would "push" approval requests. The emails would have "Approve"/ "Decline" links. This would speed the process, and not require users to log in to a system to approve/decline. Third, all interfaces would be mobile-friendly. A request and approval could happen entirely on any device including small-screen mobile phones. Again, this would speed the in-process time.

Several other requirements became apparent through the analysis. Because many committee members were not affiliated with the university, the system could not use university accounts for authentication. To overcome these challenges, a decision was made to use Google Third-party authentication using OpenAuth for the initial system, with the intention of adding others in the future, e.g., Facebook or Twitter.

For ease-of-use and more natural interactivity, it was clear that we would need to use asynchronous server calls (AJAX) (for more information on AJAX, see Garret, 2005). This would eliminate the page refreshes that can be disconcerting on a wireless device with spotty service.

Two external systems were identified as possibly needing interfaces in the new system: LinkedIn and the program mailing list. The program maintains an active LinkedIn group for alumni, current students, employers, and others. All proposal and defense announcements are posted to LinkedIn in a fairly consistent format. This is normally done by the student, but the design team chose to include this to increase automation. In addition, a mailing list is used primarily for internal program announcements to faculty and current students. Again, this is normally manually done by a program administrator, but we thought that it might be automated to some extent.

Finally, the approval system would need to integrate with an existing system that published capstone/thesis documents on the web. This publishing system lists all completed thesis in list form, provides an indexer-friendly "landing page" with abstract and citation information for each document, enables social media "likes" and sharing for each document, and cross links to faculty, students, and related publications. The new system would reside on the same web server, and share the same database.

3. DESIGN

Appendix 1 shows the workflow of the system, developed through interviews with key stakeholders. It is more formal than the paper-form process, and represents some subtleties not captured before. For example, the composition of the committee requires committee chair approval before the requests go to the faculty. Of course, it was expected that students would discuss their committee selection with their chair, but there was no enforcement of this, which led to some misunderstandings under the legacy process. Note that "decline"s or time-outs at each stage in the process merely revert the student to the previous stage.

Appendix 2 is the Actor Diagram describing the users and how they interact with the system. Note that some of the use cases are initiated from within an email, and do not require a full login by the user. The four main Actors are:

- Student
- Chairperson
- Committee Member
- Director

In addition to the main Actors above, we needed a user interface for a System Administrator. This would be used to monitor and manage the technical aspects of the system. The general public is the final Actor, consuming notifications and items published in LinkedIn, the mailing list, and the annals web site.

The "Student Requests Chairperson" use case is shown in more detail as part of Appendix 3. This is typical of the use cases for this system – it is short and relatively simple. There are no complex interactions. Appendix 4 includes emails that are created via the student request for chairperson. The student initiates an approval, an email is sent to the potential chairperson, who clicks on "Accept" or "Decline" links, and the action is recorded. Status update emails are sent, and the student's status changes to "Chair Accepted".

Google acts as the third-party authenticator, using the OpenAuth protocol. In cases where a full login is required, users are presented with a Google Login dialog box. Upon successful authentication, an SSL connection is established with the server, and future interaction is encrypted. Google handles changes to passwords, password reminders, etc. Users have one less username/password to remember.

Bootstrap (Otto & Thornton, 2013) was used to style the web interfaces and provide responsive screens for devices of all sizes. Screens were primarily designed for small screens such as smart phones. For larger devices, Bootstrap elegantly expands to fill the screen and "unstack" UI components. We found that for the short use case interactions that make up most of this system, the user-preferred device was a smartphone.

AngularJS (https://angularjs.org/) was seriously evaluated and considered for use in this system. The two very attractive features of this framework were two-way binding and declarative form validation. However, the main platform of ASP.Net with C# and JavaScript met our actual needs. The user interfaces were not complex enough to require two-way binding, and the ASP.Net controls provided declarative form validation without round-trips to the server. In the end, we felt that the potential benefits of AngularJS did not outweigh the complexity of an additional framework.

The general paradigm for this system was a Single-Page Application (SPA). Technically, there are multiple pages – one for each main Actor, and for interactions requiring an initial page load (below):

- Default.aspx
- Director.aspx
- Student.aspx
- AuthenticationEndpoint.aspx
- EmailResponse.aspx

This was mainly done to organize the code base. However, the Default, Director, and Student pages could easily have been combined. All interactions within these pages were exclusively done by asynchronous JavaScript calls to the server (AJAX).

The database schema is shown in Appendix 5. Most of the tables already existed to support the document publishing system for the annals. The five tables in the lower right of the diagram were added to support the approval system. The CapstoneActivity table acts as a log, recording every action associated with every capstone. The CapstoneStatus table keeps the major status changes represented in the workflow. The CapstoneAdminConfiguration table holds system configuration information, such as the email templates, which can be modified without recompiling the system. CapstoneAction and Status hold the valid values for the related tables. In large part, the workflow and operation of the system could be modified entirely by changing table entries without changes to the code base.

An example of various screens the student interacts with is included in Appendix 6. These screens exemplify the relatively simple use cases that make up the bulk of the interactions with this system. They fit well on a small screen, and have the look-and-feel of a native application. The Request Committee Members screen represents a more complex interaction, allowing students to add committee members not already in the database, e.g., experts from the professional community. This interaction still works very well on a small screen with touch-screen data entry.

Appendix 7 shows the main status page for a student. On the left are the steps that must be completed (from top to bottom), with a green checkmark indicating completion. This screenshot represents a student who has completed the entire process. The progress bar shows percentage complete for the entire process. The center panels give summary information about the capstone. The right panel shows the log of all activity associated with this capstone.

This screen solved a significant problem for the program. Students only go through this process once in their life, and were therefore not familiar with the sequence of activities. This system lays out the process clearly from top to bottom, and forces a "lock-step" navigation through the process. Students found this extremely helpful.

The new system also includes a program director's view into the data showing the status of all capstones (in-process and completed). In the past, program-level reports were manual, and difficult to create and update. The new view shows all completed capstone and thesis documents for the entire program. This includes various tabs along the top of the screen providing program-level reports for "Upcoming Proposals", etc. These reports are always available and represent real-time status of students' progress.

The administrator user interface (not shown) includes screens for entering users (students and faculty), and manually recording or modifying information.

The overall architecture of the system is shown in Appendix 8. The entire system is broken into three main layers: User Interface, Business Logic, and Data Access. The User Interface has surprisingly few pages compared with traditional web-form development. The Business Logic layer consists of utility classes that are loaded on application startup, configuration, and pages to support Google OpenAuth Federated login.

The Data Access layer consists of datasets XML Entity Framework datasets, and sixty-eight (68) stored procedures. The stored procedures reside on a MS SQL Server database server and are written in Transact-SQL. Each one is very specific, and performs minimal database operations. Transactional consistency is enforced with BEGIN TRAN, COMMIT TRAN, and ROLLBACK TRAN directives, and appropriate transaction isolation levels.

4. CONCLUSION/DISCUSSION

The system was well received by both students and faculty. While presenting the system, the entire workflow was completed by a student and committee in about 5 minutes, with all participants on smart phones. The participants were not trained ahead of time, and had no prior experience with the system. This is compared to the average time currently to complete just one step in the process which usually takes 1 to 2 weeks. The goal of the system is to streamline the process and centralize all paperwork/process steps to eliminate loss of signed forms.

Most of the design goals were met. Physical signatures were eliminated for all forms except the Proposal and Defense approvals. These two forms represent degree program requirements, which merited more formal physical requirements. These two forms are the most easily completed, since they are generally signed at the end of the proposal and defense, when the committee is all in one room.

Third-party authentication was accomplished through Google to support participants from outside the university. This was surprisingly simple, and required minimal code. Authentication is generally a difficult part of a system and would have taken significant time and effort if it was incorporated into the scope of the project.

Overall, the look-and-feel of the final system was good and user-friendly. The out-of-the-box Bootstrap styles met most of the needs of the system with very few changes being made to these styles. On a smartphone, the pages feel like native applications and are very responsive. Receiving an email request, clicking on an approval link, and seeing the confirmation screen is smooth and easy.

The design goals of integration with LinkedIn and the mailing list was limited, mainly due to time and resource limitations. Students still had to enter their announcement on LinkedIn. They could then copy the LinkedIn announcement URL and enter it in the system. That URL could then be placed on web pages and used by program administrators to send to the mailing list. Even with this limited support for the external systems, all parties saved time, and the process was more consistent.

Our main concerns for this project were time and resources. The system used cutting edge features which were new to the developer. Web-based systems are notorious for having many technologies that must work together, and many languages (JavaScript, JQuery, Bootstrap, C#, ADO.Net, SQL, T-SQL). However, the final architecture was relatively simple and elegant. Detailed, comprehensive systems analysis and design phases were extremely helpful in reducing the overall time. The implementation phase ran smoothly without significant design changes.

Possible future enhancement for the system include:

- Text notifications
- More third-party authentication support: Facebook, LinkedIn, Twitter, etc.
- File uploads
- Extension to other graduate programs at the university
- Graphical reports

Text notifications would require a Short Message Server (SMS) capability, typically through a third-party provider such as Google. Third-party authentication extension is relatively easy, but requires developer accounts, libraries, and configuration for each provider.

Providing the ability for students to upload the final document, and request approvals for it would truly complete the process. This would eliminate large email attachments and "lost in the email inbox" issues. Furthermore, documents could be instantly published to the annals website without manual processes.

From the program director's viewpoint, more aggregate reporting would be immensely helpful. For example, a pie-chart representing the current student body's status would show the "pipeline", and help with enrollment management. Time-series style graphs could show graduations over time. Analytics could show, for example, the average time-in-process by committee chairperson.

Overall, the benefits of the system are significant. Students have a clear view of the process, and are relieved of the need to track down faculty for signatures. Faculty and administrators can see the progress of students. Program-level reporting is available for better decision-making.

4. REFERENCES

Bruhn, R.E. & Camp, J. (2004). Capstone Course Creates Useful Business Products and Corporate-Ready Students. ACM SIGCSE Bulletin, 36(2), 87-92.

Garrett, Jesse James. (2005) Ajax: A new approach to web applications. Retrieved June 19, 2015 from https://courses.cs.washington.edu/courses/cse490h/07sp/readings/ajax_adaptive_path.pdf

Goold, A. (2003). Providing Process for Projects in Capstone Courses. Proceedings of the 8th Annual Conference on Innovation and Technology in CS Education, ACM SIGCSE Bulletin, 35(3), 26-29.

Kline, D.M, Vetter, R., Barnhill, K. (2012) Building an Effective Interdisciplinary Professional Master's Degree. Information Systems Education Journal, 11(6) pp 40-49.

Novitzki, J.E. (2001). Critical Issues in the Administration of an Integrating Capstone Course. Proceedings Informing Science and Information Technology Education 2001, 372-378.

Olarte, J. J., Dominguez, C., Jaime, A., & Garcia-Izquierdo, F. J. (2014). A tool for capstone project management in computer science engineering. In *Computers in Education (SIIE), 2014 International Symposium on,* 65-68.

Otto, M., & Thornton, J. (2013) "Bootstrap." Bootstrap. http://getbootstrap.com/

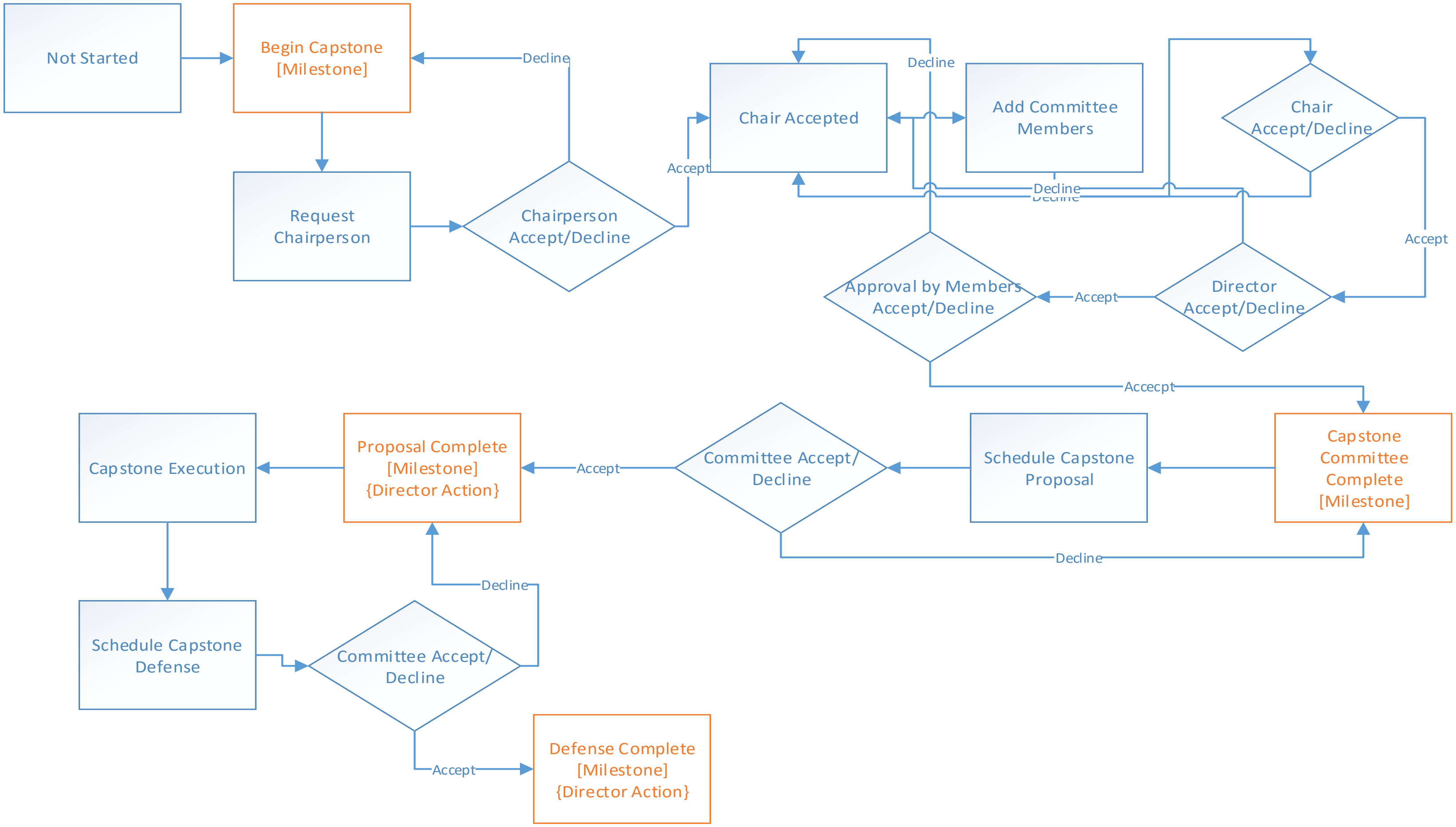
Pérez, C. D., Elizondo, A. J., Garcia-Izquierdo, F. J., & Larrea, J. J. O. (2012). Supervision typology in computer science engineering capstone projects. Journal of Engineering Education, 101(4), 679-697.

Reinicke, B., Janicki, T. (2011). Real World Projects, Real World Problems: Capstones for External Clients *Information Systems Education Journal,* 9(3), 23-27

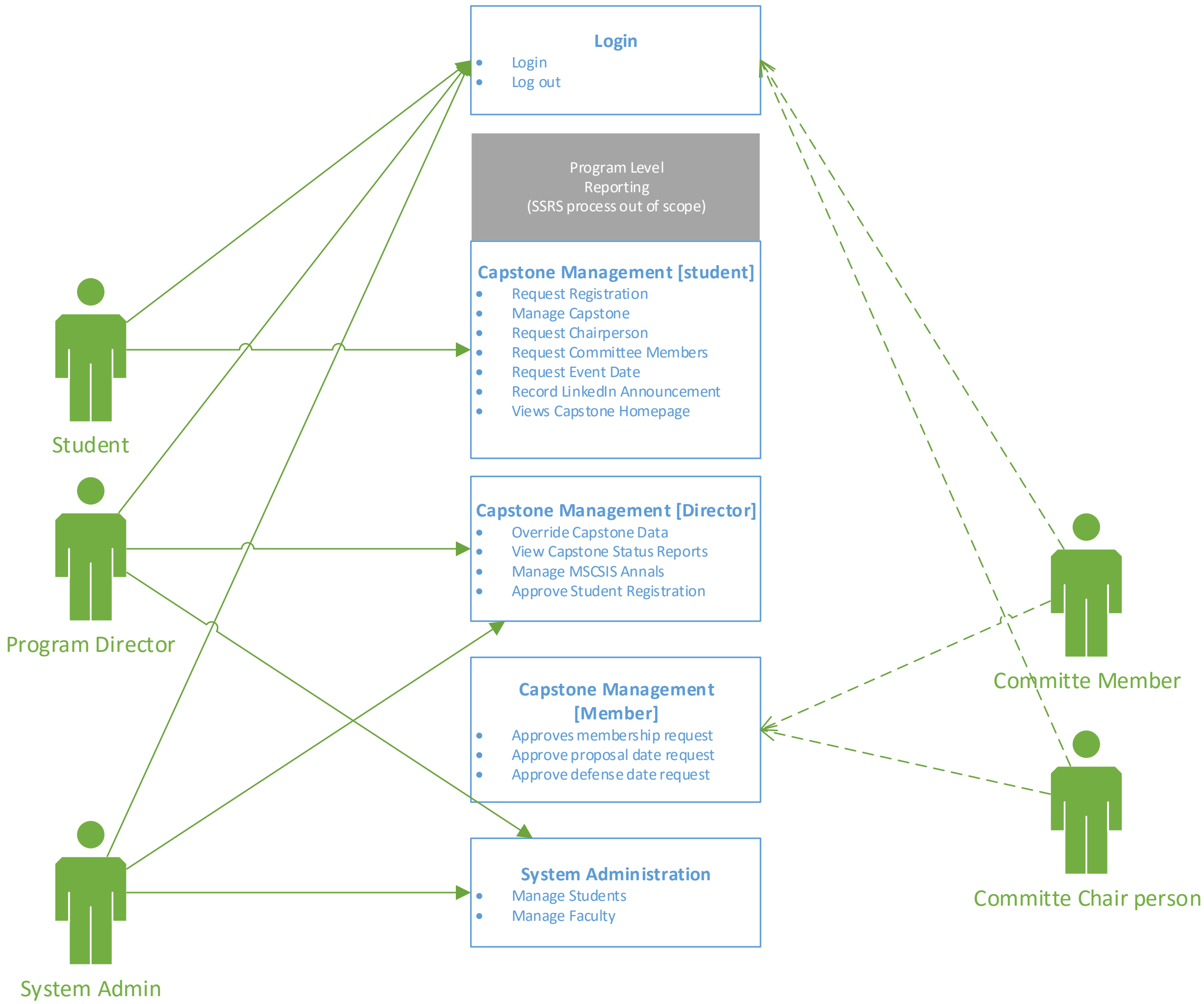
Schwieger, D., Surendran, K. (2011). Incorporating Capstone Courses in Programs Based on the IS2010 Model Curriculum. *Information Systems Education Journal,* 9(2), 65-74.

W3 Schools (2014) *AJAX Introduction*. http://www.w3schools.com/ajax/ajax_intro.asp

APPENDIX 1. CAPSTONE MANGEMENT SYSTEM WORKFLOW



APPENDIX 2. CAPSTONE MANGEMENT CAPSTONE DIAGRAM



APPENDIX 3. CHAIRPOERSON REQUEST USE CASE

Use Case	**Student Requests Chairperson**	
Description	The student requests a faculty member as the committee chairperson	
Frequency	Episodic: 10-30/semester	
Actors	Student	
Related Use Cases	Login, Log Out	
Stakeholders	Student, Committee Member	
Happy Pathway	Student selects faculty member as chairperson and notification goes out	
Preconditions	User does not have a committee chairperson	
Post-Conditions	Committee chairperson is emailed a request	
Flow of Events	**Actor**	**System**
	1. Student clicks Request Committee Chair from status panel 2. Student selects chair person 3. Student clicks save button	4. Email notification is sent to selected user 5. Action recorded
Alternate Paths	N/A	
Exception Conditions	System Timeout. No data is stored	

APPENDIX 4 – CHAIRPERSON REQUEST SYSTEM EMAILS-

Committee Chairperson Request

To: Faculty Member

Subject: {Student Name} Capstone Chairperson Request

Hello,

Please consider becoming my chairperson for my capstone committee.

Title: {Capstone Title}

Click this link to accept: I accept

Click this link to decline: I decline

Click Here to login

This is a system generated email. Please do not reply.

Committee Chairperson Response (Declined)

To: Student

Subject: Capstone Committee Request Response

Your capstone committee chairperson request has been declined by {Chairperson Name}. Please login to the capstone management system to continue.

Click Here to login

This is a system generated email. Please do not reply.

Committee Chairperson Response (Accepted)

To: Student

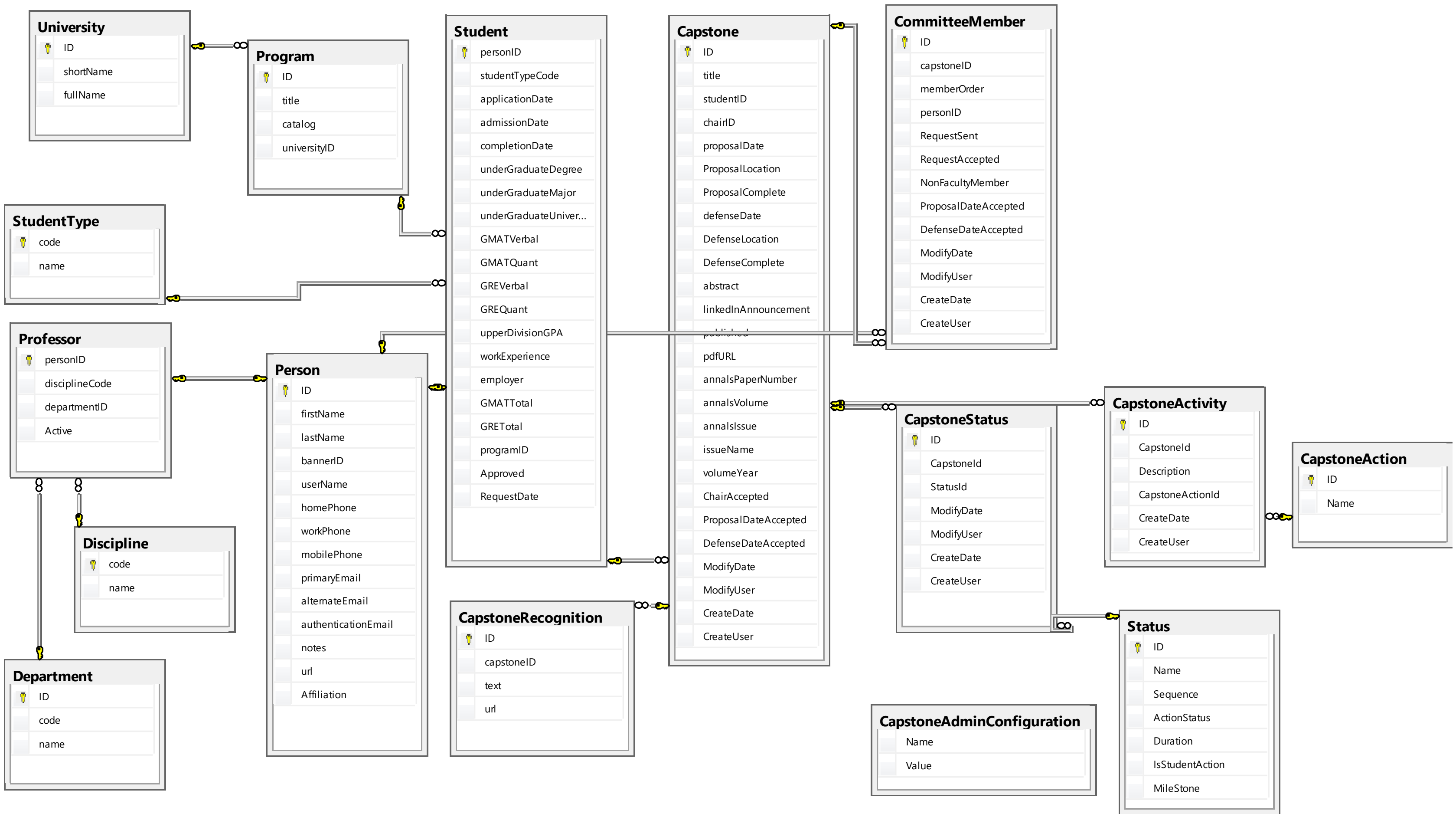
Subject: Capstone Committee Request Response

Congratulations! Your capstone committee chairperson request has been accepted by {Chairperson Name}. Please login to the capstone management system to continue.

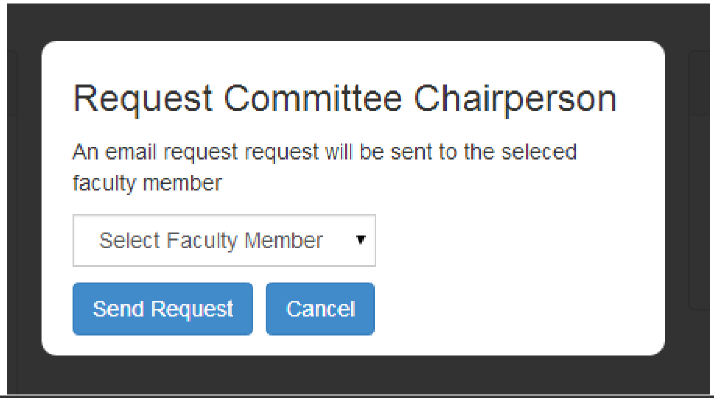
Click Here to login

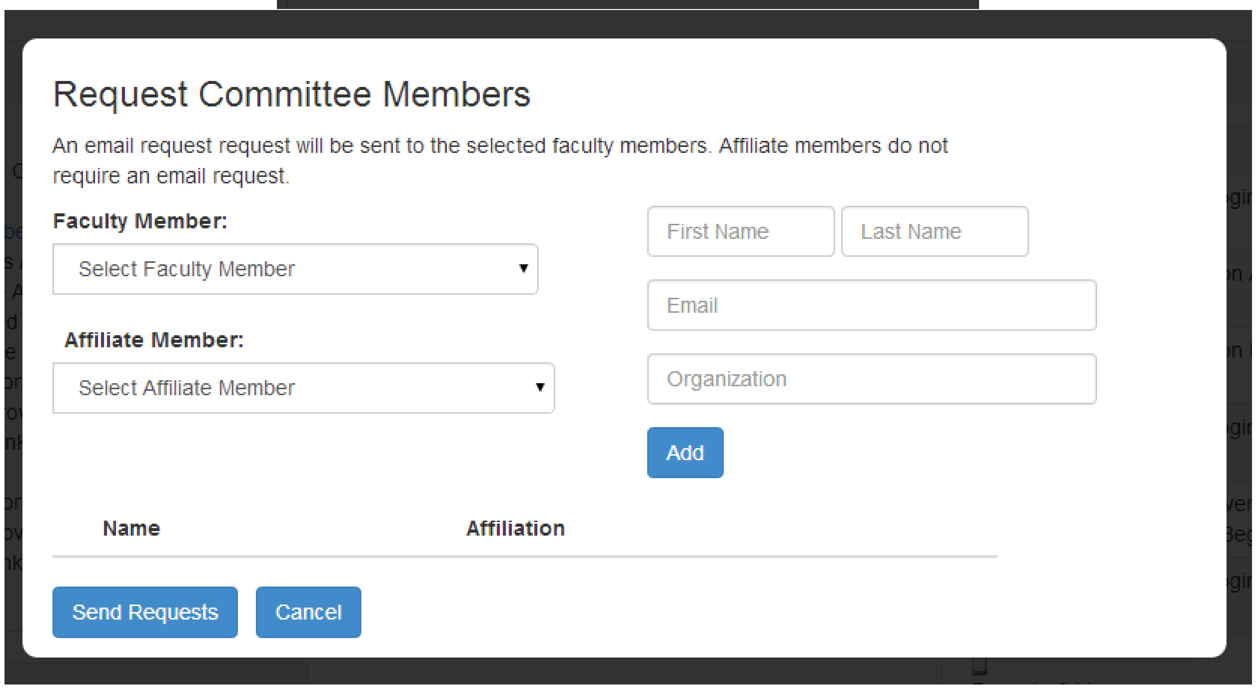
This is a system generated email. Please do not reply.

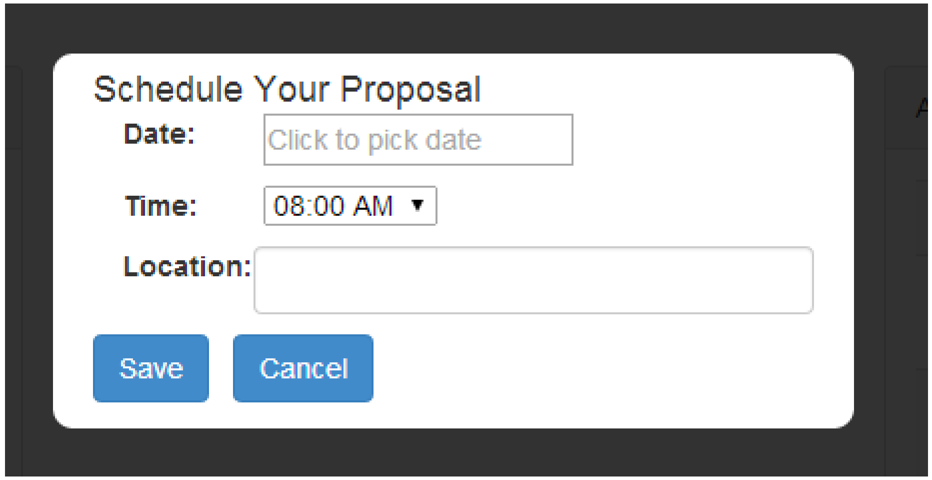
APPENDIX 5. CAPSTONE MANAGEMENT DATABASE SCHEMA



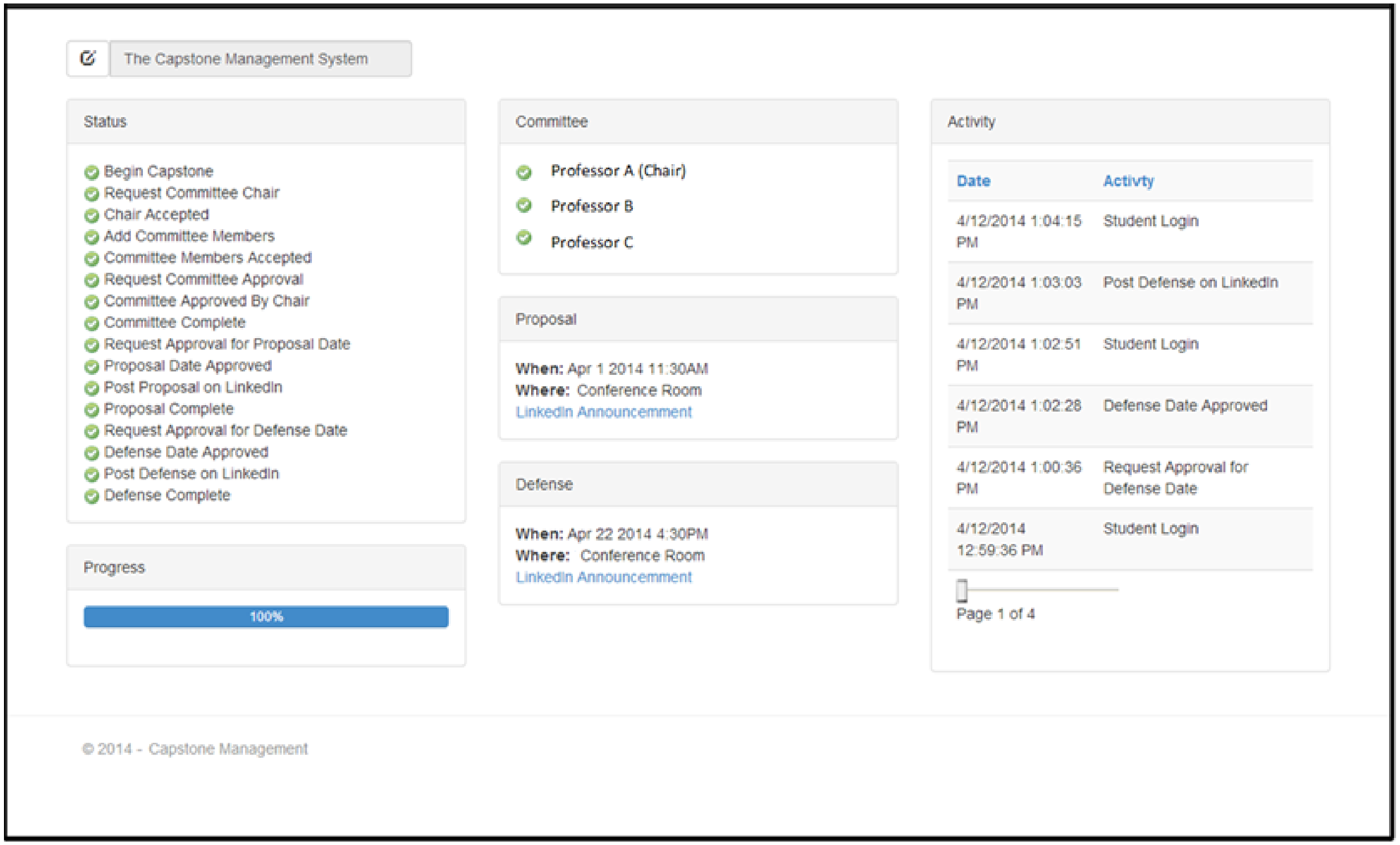
APPENDIX 5. CAPSTONE MANAGEMENT DATABASE SCHEMA





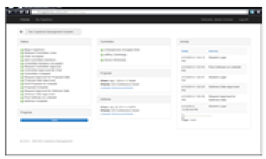

Schedule Your Proposal
Date:
Click to pick date
Time:
08:00 AM
Location:
Save
Cancel

APPENDIX 7. STUDENT STATUS HOMEPAGE



APPENDIX 8. CAPSTONE MANAGEMENT SYSTEM ARCHITECTURE

User Interface / Application



ASPX Pages:
Default.aspx/Default.aspx.cs
Student.aspx/Student.aspx.cs
Director.aspx/Director.aspx.cs
Adminisrtrator.aspx/Administrator.aspx.cs
EmailResponse.aspx/EmailResponse.aspx.cs
Home.aspx/Home.aspx.cs
Site.Master/Site.Master.cs

Bootstrap UI files:
Content/bootstrap.min.css
Content/bootstrap-theme.min.css
Content/modal.css
Content/Site.css
Script/bootstrp.min.js
Script/jquery-1.10.2.min.js
Script/modernizr-2.6.2.js
Script/respond.min.js

Business Logic Layer

CSB_APP

Utility Classes:
App_Start/CapstoneEmail.cs
App_Start/CapstoneCommitee.cs
App_Start/CapstoneStatus.cs
App_Start/Activity.cs

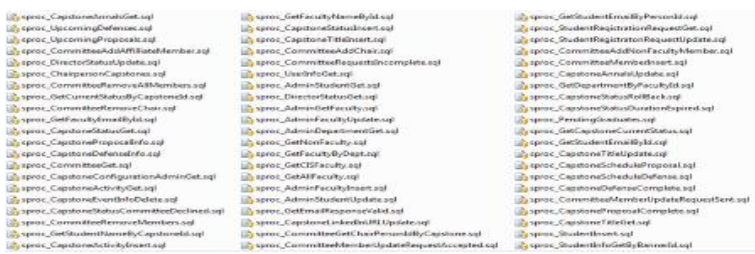
Global Enumerations / Settings:
App_Start/AppConfig.cs
Web.config

Federated Login:
AuthenticatonEndpoint.aspx
AuthenticatonEndpoint.aspx.cs

Data Access Layer

Datasets:
Activity.xsd
Adminitrator.xsd
Capstone.xsd
Committee.xsd
Config.xsd

Stored Procedures


sproc_CapstoneAnnalsGet.sql
sproc_UpcomingDefenses.sql
sproc_UpcomingProposals.sql
sproc_CommitteeAddAffiliateMember.sql
sproc_DirectorStatusUpdate.sql
sproc_ChairpersonCapstones.sql
sproc_CommitteeRemoveAllMembers.sql
sproc_GetCurrentStatusByCapstoneId.sql
sproc_CommitteeRemoveChair.sql
sproc_GetFacultyEmailById.sql
sproc_CapstoneStatusGet.sql
sproc_CapstoneProposalInfo.sql
sproc_CapstoneDefenseInfo.sql
sproc_CommitteeGet.sql
sproc_CapstoneConfigurationAdminGet.sql
sproc_CapstoneActivityGet.sql
sproc_CapstoneEventInfoDelete.sql
sproc_CapstoneStatusCommitteeDeclined.sql
sproc_CommitteeRemoveMembers.sql
sproc_GetStudentNameByCapstoneId.sql
sproc_CapstoneActivityInsert.sql
sproc_GetFacultyNameById.sql
sproc_CapstoneStatusInsert.sql
sproc_CapstoneTitleInsert.sql
sproc_CommitteeAddChair.sql
sproc_CommitteeRequestsIncomplete.sql
sproc_UserInfoGet.sql
sproc_AdminStudentGet.sql
sproc_DirectorStatusGet.sql
sproc_AdminGetFaculty.sql
sproc_AdminFacultyUpdate.sql
sproc_AdminDepartmentGet.sql
sproc_GetNonFaculty.sql
sproc_GetFacultyByDept.sql
sproc_GetCISFaculty.sql
sproc_GetAllFaculty.sql
sproc_AdminFacultyInsert.sql
sproc_AdminStudentUpdate.sql
sproc_GetEmailResponseValid.sql
sproc_CapstoneLinkedInURLUpdate.sql
sproc_CommitteeGetChairPersonIdByCapstone.sql
sproc_CommitteeMemberUpdateRequestAccepted.sql
sproc_GetStudentEmailByPersonId.sql
sproc_StudentRegistrationRequestGet.sql
sproc_StudentRegistratonRequestUpdate.sql
sproc_CommitteeAddNonFacultyMember.sql
sproc_CommitteeMemberInsert.sql
sproc_CapstoneAnnalsUpdate.sql
sproc_GetDepartmentByFacultyId.sql
sproc_CapstoneStatusRollBack.sql
sproc_CapstoneStatusDurationExpired.sql
sproc_PendingGraduates.sql
sproc_GetCapstoneCurrentStatus.sql
sproc_GetStudentEmailById.sql
sproc_CapstoneTitleUpdate.sql
sproc_CapstoneScheduleProposal.sql
sproc_CapstoneScheduleDefense.sql
sproc_CapstoneDefenseComplete.sql
sproc_CommitteeMemberUpdateRequestSent.sql
sproc_CapstoneProposalComplete.sql
sproc_CapstoneTitleGet.sql
sproc_StudentInsert.sql
sproc_StudentInfoGetByBannerId.sql