



ISSN: 1545-679X

Information Systems Education Journal

Volume 8, Number 40

<http://isedj.org/8/40/>

June 30, 2010

In this issue:

Increasing Active Learning and End-Client Interaction in the Systems Analysis and Design and Capstone Courses

Bryan A. Reinicke

University of North Carolina Wilmington
Wilmington, NC 28403 USA

Thomas N. Janicki

University of North Carolina Wilmington
Wilmington, NC 28403 USA

Abstract: Systems analysis and design (SAD) is one of the core courses offered in most IS programs, yet this class can be challenging for students and instructors alike. The concepts can be abstract, and getting students to appreciate their importance can be difficult. This paper discusses the implementation of a two semester sequence in which the students are placed in teams to complete an analysis, design and implementation of a real world project for an end client. The result is the theories of the systems analysis and design course are placed into practice immediately through active learning and also the capstone projects have a higher level of success.

Keywords: Systems Analysis and Design, capstone course, active learning, service learning, pedagogy

Recommended Citation: Reinicke and Janicki (2010). Increasing Active Learning and End-Client Interaction in the Systems Analysis and Design and Capstone Courses. *Information Systems Education Journal*, 8 (40). <http://isedj.org/8/40/>. ISSN: 1545-679X. (A preliminary version appears in *The Proceedings of ISECON 2009*: §2356. ISSN: 1542-7382.)

This issue is on the Internet at <http://isedj.org/8/40/>

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: <http://isedj.org/> • Contact person: Don Colton (editor@isedj.org)

2010 AITP Education Special Interest Group Board of Directors

Don Colton Brigham Young Univ Hawaii EDSIG President 2007-2008	Thomas N. Janicki Univ NC Wilmington EDSIG President 2009-2010	Alan R. Peslak Penn State Vice President 2010	
Scott Hunsinger Appalachian State Membership 2010	Michael A. Smith High Point Univ Secretary 2010	Brenda McAleer U Maine Augusta Treasurer 2010	George S. Nezelek Grand Valley State Director 2009-2010
Patricia Sendall Merrimack College Director 2009-2010	Li-Jen Shannon Sam Houston State Director 2009-2010	Michael Battig St Michael's College Director 2010-2011	Mary Lind North Carolina A&T Director 2010-2011
Albert L. Harris Appalachian St JISE Editor ret.	S. E. Kruck James Madison U JISE Editor	Wendy Ceccucci Quinnipiac University Conferences Chair 2010	Kevin Jetton Texas State FITE Liaison 2010

Information Systems Education Journal Editors

Don Colton Professor BYU Hawaii Editor	Thomas N. Janicki Associate Professor Univ NC Wilmington Associate Editor	Alan R. Peslak Associate Professor Penn State Univ Associate Editor	Scott Hunsinger Assistant Professor Appalachian State Associate Editor
---	--	--	---

Information Systems Education Journal 2009-2010 Editorial and Review Board

Samuel Abraham, Siena Heights	Brenda McAleer, U Maine Augusta	Mark Segall, Metropolitan S Denver
Alan Abrahams, Virginia Tech	Fortune Mhlanga, Abilene Christian	Patricia Sendall, Merrimack Coll
Ronald Babin, Ryerson Univ	George Nezelek, Grand Valley St U	Li-Jen Shannon, Sam Houston St
Michael Battig, St Michael's C	Anene L. Nnolim, Lawrence Tech	Michael Smith, High Point Univ
Eric Breimer, Siena College	Monica Parzinger, St Mary's Univ	Robert Sweeney, South Alabama
Gerald DeHondt II, Grand Valley	Don Petkov, E Conn State Univ	Karthikeyan Umamathy, U N Florida
Janet Helwig, Dominican Univ	Steve Reames, American Univ BIH	Stuart Varden, Pace University
Mark Jones, Lock Haven Univ	Jack Russell, Northwestern St U	Laurie Werner, Miami University
Terri Lenox, Westminster Coll	Sam Sambasivam, Azusa Pacific U	Bruce A. White, Quinnipiac Univ
Mary Lind, NC A&T University	Bruce M. Saulnier, Quinnipiac	Charles Woratschek, Robert Morris
Cynthia Martincic, St Vincent C		Peter Y. Wu, Robert Morris Univ

This paper was in the 2009 cohort from which the top 45% were accepted for journal publication. Acceptance is competitive based on at least three double-blind peer reviews plus additional single-blind reviews by the review board and editors to assess final manuscript quality including the importance of what was said and the clarity of presentation.

© Copyright 2010 EDSIG. In the spirit of academic freedom, permission is granted to make and distribute unlimited copies of this issue in its PDF or printed form, so long as the entire document is presented, and it is not modified in any substantial way.

Increasing Active Learning and End-Client Interaction in the Systems Analysis and Design and Capstone Courses

Bryan Reinicke
reinickeb@uncw.edu

Thomas Janicki
janickit@uncw.edu

University of North Carolina Wilmington
Information Systems and Operations Management
Cameron School of Business
Wilmington, North Carolina 28403 USA

Abstract

Systems analysis and design (SAD) is one of the core courses offered in most IS programs, yet this class can be challenging for students and instructors alike. The concepts can be abstract, and getting students to appreciate their importance can be difficult. This paper discusses the implementation of a two semester sequence in which the students are placed in teams to complete an analysis, design and implementation of a real world project for an end client. The result is the theories of the systems analysis and design course are placed into practice immediately through active learning and also the capstone projects have a higher level of success.

Keywords: Systems Analysis and Design, capstone course, active learning, service learning, pedagogy

1. INTRODUCTION

Systems analysis and design (SAD) is one of the core courses offered in most IS programs, yet this class can be challenging for students and instructors alike. The concepts can be abstract, and getting students to appreciate their importance can be difficult. Chen (Chen, 2006) notes that teaching SA&D courses are more difficult than other IS courses which are more structured such as programming and database. One way to improve this situation is to involve the students in real systems analysis and design projects, but this again only captures a portion of the problem. If the students don't have to build the systems they design, many of the nuances of design may escape them.

Of course, it is also very difficult to have the students learn how to design and then build a system in a single semester.

In order to overcome this limitation, the authors have linked the systems analysis and a separate systems implementation / capstone course. Over a period of two semesters the students not only learn the conceptual frameworks for systems analysis and design, but they must actually complete a real world analysis and design project and then implement that project for a real world client. This paper will explore the historical structure of the SAD and capstone courses, the methods used to integrate them, and will show how this links to both the IS model curriculum, the goals of the University and can improve student learning in these two areas.

2. LEARNING THEORIES

There are various learning theories that all suggest that students need to be more involved in their learning through actual experiences. The learning theories from the behavior, cognitive, constructionist, resource based and active learning theories all suggest methods to engage the student to be more involved in their learning. Gagne, Briggs & Wager (Gagne, Briggs, & Wager, 1988), in their study of behavior and cognitive learning theories, proposed several principles for effective instructional design that are founded in behavioral learning theory. One of these principles is contiguity, the concept that the response should follow the stimulus without delay. The longer the delay of the response to a learning stimulus, the less likely the student is to retain the learning. Thus if the students can practice what they have learned, learning may be increased.

Likewise Brandt (1997) observes that learners construct knowledge by making sense of experiences in terms of what is already known. Learners transfer knowledge through experiences via mental models, which are used to assimilate new information into knowledge, and thus become expanded mental models. This knowledge transfer (defined as constructivism (Brandt, 1997)), emphasizes knowledge construction and problem solving in domains of increasing conceptual complexity.

Traditional Learning	Resource-Based Learning
Teacher as an expert model	Teacher as a facilitator/guide
Textbook as primary source	Variety of sources/media
Facts as primary	Questions as primary
Information is packaged	Information is discovered
Emphasis on product	Emphasis on process
Assessment is quantitative	Assessment is qualitative/quantitative

Table 1: Traditional versus Resource Based Learning (Rakes, 1996)

The resource view of learning depicts a changing role for the instructor, from that of

an expert dispensing knowledge to one of a resource and a guide. Rakes (Rakes, 1996) envisions a change from traditional learning to one based on a multitude of resources being made available to a student. Rakes (1996) supports the move to 'resource based learning.' He recommends increasing a student's success through the addition of practice to shift from the traditional view of learning (cognitive and behavioral) to a resource-based view of learning. Table 1 provides a comparison of the traditional view and resource view of learning.

Chickering and Gamson (Chickering & Gamson, 1999) discuss the need for 'active learning,' in which they state that students do not learn just by sitting in class listening to teachers, but that they need to be involved in the process and make it part of their personal experiences. Bonwell and Eison (Bonwell & Eison, 1991) also state that students must engage in higher thinking tasks such as analysis, synthesis and evaluation, in order to better incorporate the learning into their experiences. We have used these concepts as the basis for the idea that students will learn the analysis and design concepts if they must employ them as part of their learning experience.

3. BACKGROUND

Systems Analysis and Design

Systems analysis and design is one of the key courses for the information systems curriculum. The IS 2009 Model Curriculum "draft" (IS2009, 2009) details that systems analysis and design should be one of the key courses for all IS majors. One of the requirements is that IS students should exhibit strong "analytical and critical thinking, including creativity and ethical analysis. "Every IS professional must have strong analytical and critical thinking skills. Fundamentally, IS students need to master problem solving and systems thinking skills to analyze, design, develop, and evaluate IS systems and situations." (IS 2009, p16)

However, it can be difficult for the students to understand why the material is important. After all, for many programs, the SAD course is a "soft" skills course, while many of the other classes (database and programming courses) tend to be "hard" skills courses. It is easy to see the immediate application for the skills learned in each of these hard skills classes, as the students will

not be able to create or query the database without the appropriate SQL statements. In SAD, however, it is hard to see what the immediate application of the skills is – they are simply more difficult to operationalize.

Historically at our university (a mid sized public institution), the SAD course was taught strictly from a text book, with very few opportunities for the students to put the skills they were learning to use in the course. However, the department had implemented a separate required capstone experience course for all IS majors. In this capstone course, students were required to build a system for a client either on campus or for a local non-profit or business. This arrangement provided the opportunity to meet a number of requirements. First, it allowed the students to learn about systems hands-on in the classroom before going out into the job market. Second, it provided a benefit for the organizations (both on and off campus) that the students worked with. Third, it fulfilled a portion of the university mission (namely for service learning and regional engagement). After three semesters of the capstone course, it was observed that it was very difficult to have the students both design and build the systems in a single semester, while simultaneously learning the framework that they would use to implement it. In the course evaluations, students felt overwhelmed and did not feel they were able to implement good SAD practices (from design through construction and deployment) in this one semester capstone.

This set of problems led to discussions between the two authors, who are responsible for the SAD and capstone courses at the university. The authors discussed the possibility of having the students learn SAD and design the systems in the SAD course, which they would then be responsible for implementing in the capstone course. The SAD course is the pre-requisite for the capstone course. This seemed like an excellent opportunity to solve several problems for the courses, as well as to increase the learning of students in both courses.

As an extra incentive for increasing the SAD skill sets of our students IS graduates is the research by Woratscheck and Lenox (2002), where they surveyed employers to determine the skills they expect graduates to have. Woratscheck and Lenox (2002) reported that non-technical skills (such as those emphasized in SAD) were as impor-

tant as technical skills. In addition, their survey of 30 plus employers noted that knowledge of the systems development life cycle remains a key component of IS graduate knowledge, with less emphasis on programming languages. Janicki, et.al (2008) survey of over 300 IS employers also indicated the need for strong analysis, thinking and design skills.

IS, as a field, focuses on two primary areas, the “acquisition, deployment, and management of information technology resources and services” and the “development and evolution of technology infrastructures and systems for use in organizational processes.” (IS 2009) The inclusion of real world projects into these two courses ties directly into these goals for IS.

Capstone Course

Capstone projects are widely used in business degree programs (Payne, Flynn, & Whitfield, 2008) to provide students with the opportunity to work on a “real life” project. These projects have been shown to have benefits for both students and faculty. In general, these projects provide the opportunity for students to synthesize the knowledge they have accumulated through their courses and apply it. In addition, for projects in information systems, the capstone experience allows the student to interact with a client for whom they are developing a system – an experience which cannot be replicated from book work. This type of project has also been shown to improve the students self efficacy with problem solving (Dunlap, 2005).

These projects also tie in with a more recent trend in university education: that of service learning (Govekar & Rishi, 2007; Gujarathi & McQuade, 2002; Rose, Rose, & Norman, 2005). In addition to this being a trend in higher education generally, it is listed as one of the key missions of the host university of the authors. (UNCW, 2009) Service learning has multiple benefits for students by engaging them in the community in which they are learning, and by allowing them to develop leadership skills as they work through the project. Studies have shown that a service learning approach can also increase student satisfaction and the desire to learn (Rose et al. 2005). This type of learning experience also allows faculty to incorporate the real world into the classroom (Govekar and Rishi 2007; (Godfrey & Grasso,

2000), and to directly demonstrate why the concepts they are discussing are important.

Model Curriculum Implementation

The IS 2009 model curriculum defines the importance of designing and implementing systems solutions. The models states "Those who can demonstrate the ability to integrate high performance in design and implementation, along with strong business capabilities, are typically the most highly sought after graduation." (IS 2009 Model Curriculum p.21).

Specific recommendations for the course include the following items (IS 2009, p.49), and we also describe how the authors attempted to implement that concept.

1. Students will learn to understand the types of business needs that can be addressed using information technology-based solutions.

In the SAD course, the students discuss the types of needs that businesses have (to maintain a competitive advantage or to support their strategy, for example) and how systems can be designed to support and enable those goals.

2. Students will learn to initiate, specify, and prioritize information systems projects and to determine various aspects of feasibility of these projects.

In the SAD course, the students learn various means that organizations can use to identify promising projects and to rank order them. Specifically, the financial and strategic returns and the means to evaluate these are covered in the course. Project management concepts and methods for initiating projects within an organization are also covered.

3. Students will learn to use at least one specific methodology for analyzing a business situation (a problem or opportunity), modeling it using a formal technique, and specifying requirements for a system that enables a productive change in a way the business is conducted. Within the context of this methodology, students will learn to write clear and concise business requirements documents and convert them into technical specifications.

In the SAD course, the students have learned the traditional method for performing a systems analysis and for systems modeling and documentation. By

engaging in projects with local organizations, they are required to gather the user requirements and transform them into technical specifications.

4. Students will learn to communicate effectively with various organizational stakeholders to collect information using a variety of techniques and to convey proposed solution characteristics to them.

In the capstone course the students are required to periodically provide prototype screens and models to the end users, seek their input and learn about 'change management.' In addition the teams make formal presentations to the end clients at the end of the semester.

5. Students will learn to manage information systems projects using formal project management methods.

In the capstone course the students develop their own project plan for completing the projects to meet the contract set with the end user. In addition to a semester project plan, the plan has three week milestones in which the student teams must report their past project accomplishments and provide specific details on what tasks will be accomplished before the next milestone.

6. Students will learn to articulate various systems acquisition alternatives, including the use of packaged systems (such as ERP, CRM, SCM, etc.) and outsourced design and development resources.

The SAD course covers the various alternatives for acquiring and building systems and the various advantages and disadvantages of these methods.

7. Students will learn to systematically compare the acquisition alternatives.

The SAD course covers the various areas in which systems may be compared, including the cost, functionality and the areas of feasibility that organizations must consider when investigating new systems.

4. KEY DELIVERABLES FOR THE COURSES

Systems Analysis and Design

The following are the major deliverables for the SAD course as well as the role of the students and the instructor. The key goal is

to make the student teams responsible for their investigation and design of the systems.

1. Determine the user needs for the system they are to develop.
 - a. Each team is required to develop a set of questions and then interview the user to get a baseline for the system project.
 - b. Based on these initial interviews, the teams develop a project charter, which is signed between both the team members and the client.
2. Follow up with users on the design
 - a. Throughout the semester the teams are required to follow up with the client when they encounter an area of requirements that are unclear. This helps to drive home the point that the collection of requirements, and system design itself, are iterative processes.
3. Create a design for the system.
 - a. The teams are each responsible for creating a logical design, and the structured business logic, for the system.
 - b. The teams also need to create mock-ups of the user screens to demonstrate how the system would work for the user.
4. Create a database design to support the system.
 - a. The teams need to create an Entity Relationship Diagram for all of the data that the system will need to contain.
5. Client Communication.
 - a. While the instructors of the SAD and Capstone course make the initial contact with clients to find the initial projects, the student teams are responsible for all communication with the client beyond this. Instructors are kept in the communications loop through cc:'s on e-mails.

Capstone Course (System Implementation / n-Tier development / end client interaction)

The following are the major deliverables for the capstone course as well as the role of the students and the instructor. The key goal is to make the student teams responsible for their planning and implementation.

1. Develop a project plan to meet the user needs. This plan must detail milestones every three weeks, what is to be accomplished and what team member is responsible for that milestone.
2. Implementation of a SQL database on a server to meet user requirements.
 - a. This reinforces what the student has learned in a prior database course, but also combines the SAD concepts for ERD's and DFD's. The database must support all user requirements
3. Build an n-tier solution for the client starting with the presentation layer
 - a. The presentation layer is built via a web interface using modern CSS (Cascading Style Sheet) concepts as well as HCI (Human Computer Interaction) concepts.
 - b. Learn the .NET visual development tools
4. Data Access Layer
 - a. Build the necessary stored procedures, authorization layers, and data base handling to support the project. Included are all CRUD (Create, Read, Update and Delete) stored procedures to reduce the risk of data injection.
 - b. Insure that updates to the data from the web pages are done based on user roles. Different classes of users have different update privileges.
5. Business Logic Layer
 - a. Learn and implement the appropriate .Net code to support the business logic requirements of the project
6. Client Communications
 - a. Meet with the client periodically to review the presentation layer screens

- b. Review periodically with the end user to insure system is working toward final goals
7. Instructor Communications
- a. Each team has a personal consultation with the instructor every three weeks. This meeting occurs after each team has updated their project plans. These plans are updated to reflect what was done (or not done) as well as revise projected targets.

5. OUTCOMES AND CONCLUSIONS

Making these changes has had a number of benefits for the learning outcomes of students. First and foremost is the experience of interviewing a "real live" user. This forces them to think of the questions they need to ask, as well as focusing them on what to do when they get less than specific answers from the users. As the users are frequently people who know that a system would make their life easier, but don't know exactly how, the students have to figure out the details.

The students also gain experience with designing a database to support a business process. This process forces the students to think about how the data should be stored without having a solution file to consult. This has proven to be more difficult for the students than many originally thought it would be which has also lead to some changes to the course design for the initial SAD course (more on this later).

Along with the database, the students have to map out a business process. This has also been challenging for the students, as it requires an eye towards detail. While there are processes in the book that the student do practice on, these examples always have gaps in them that the students then have to make assumptions about. When dealing with real users, the students are forced to go back and talk to the user to find out what the answer for the gaps is. This process is helpful, in that the students learn the level of detail required to map a business process, and that the users aren't always clear on the details!

The students also show more motivation to complete the projects with a high level of quality when they have to present a working product to a client. The students know that they will be working on the project for this client for a full year, and will be turning it

over to them at the end of the year. If it doesn't work, they have to tell the client why, and they show a reluctance to do this.

Finally, the students work in teams for these projects for a full year. This increases the students' ability and skills in group work. They have to determine who should work on which parts of the design and build, which forces them to analyze the strengths and weaknesses of each group member. They are also required to present their work to their peers on two occasions: once at the end of the SAD course when they present their final design and once at the end of Capstone Course when they present their completed project.

A key benefit of the two semester sequence is that more projects are 'production ready.' In the past when all the interviewing, analysis and design was attempted in one semester only one third of the projects would be ready at the end of the term. Since implementation of this new process, 65% of the projects have been implemented at the end of both terms.

Students also learn from each other. As there are on average 15 real world projects being developed in a semester, the concepts are very similar (authenticate a user, build menus, update data etc) and students from different teams have helped other team members. This really shows how a transfer of knowledge can occur. In 2008 we logged over 2000 service hours to the community between the two courses.

A lesson well learned from the student perspective is that clients don't always tell you what they want (or they don't know what they want). Many students comment throughout the semester, that they didn't tell us they didn't want that. This gives the instructors great opportunity to ask them to pull out their interview notes and determine if they asked the questions specific enough.

Another well learned lesson is that students often want to build their own system for their own club, church or personal activity. We have found these projects have always ended in failure. It is hard for them to identify their system needs or their own club needs.

Clients also must be managed, their expectations of 'new' developers needs to be moderated. We have found clients expect immediate production grade system very

quickly. We have done our best to indicate to the clients that this is a two semester project, their involvement in learning is most desired as well as the more time they give our students the higher quality of the projects. We have found our corporate advisory board to be an excellent source of projects as they enjoy working with our students and know they must assist in the 'training'.

These efforts also link into the schools Assurance of Learning (AOL) efforts for AACSB accreditation. The AACSB guidelines encourage the use of capstone courses for integration of knowledge and also encourage schools of business to be active in the communities they are in. This program helps to meet both of these criteria.

6. FUTURE ENHANCEMENTS

System Analysis and Design

Based on the experiences with "chaining" the two courses, several enhancements are planned for the SAD class. First among these is a change to an Object Oriented design from the traditional methods. The students frequently have trouble grasping the traditional design methods, while Object Oriented design methods follow a more "natural" path and, based on the authors' experience, students seem to have an easier time learning and applying the concepts. This will also allow the capstone students to more naturally follow an object oriented methodology, and ties in with the programming principles that are covered in other courses within the IS major. Key objects such as person, product, invoices, donations, etc. will come more naturally to the students following their initial object orientation.

A second enhancement will be requiring additional client interaction. Many of the student groups have fallen into a "traditional" pattern with system development – they meet with the client, go away and return with a "completed" system. This causes numerous problems in the real world, as well as in capstone course when the students try to build their "completed" designs, only to find out that they have missed multiple system requirements. In future semesters, the student groups will be required to meet with their clients on a regular basis and present their plans in language that the users can understand. Again, the object oriented methods will help with this, as they focus on

design standards that can be more readily understood by the users.

Another enhancement for the class will be having the students focus their design efforts on user menus as well as additional storyboards first. This will require them to list all of the user functions up front, so that fewer requirements are missed. This will also give the students something that they can share with the users that the users will be easily able to understand.

Finally, the course will spend more time on the database designs to support the systems. In the first few iterations of the course, less time was spent on this topic because the students were required to take a database course as a pre-requisite for the SAD course. However, it has become clear that the application of database design concepts continue to pose problems for the students. As such, the course will focus on this area more intensively.

Capstone Course

As the capstone course is in its fifth year and has become easier and more effective with the 'chaining' of the two courses, the following are scheduled improvements for the next academic year.

Require the students to meet with the end client more frequently. This builds a good foundation for understanding and clearing up any 'unspecified' requirements. In addition this is a great learning experience to work and interview with potential employers. Encourage our students to use these contacts in their job search.

Build more 'mini projects' to practice common requirements of all systems. This includes building common objects such as person and invoice. IN addition projects that help students create the data access layer and business logic layer for user roles will be added earlier in the semester. As the course is managing an average of 15 projects there are common features that need to be developed early and shared.

Have the student teams present their progress to the entire class at mid-terms. This will aid in students helping students. Again the concepts on many unique projects are actually similar.

REFERENCES

- Bonwell, CC, & Eison, JA. (1991). Active Learning: Creating Excitement in the Classroom.
- Chen. (2006). " Teaching Systems Analysis and Design: Bringing the Real World into the Classroom". Information Systems Education Journal, 4(84).
- Chickering, & Gamson. (1999). Development and Adaptations of the Seven Principles for Good practice in Undergraduate Education (Vol. 80).
- Dunlap, J. (2005). " Problem-Based Learning and Self-Efficacy: How a Capstone Course Prepares Students for a Profession". ETR&D, 53(1), 65-85.
- Gagne, R, Briggs, L, & Wager, W. (1988). Principles of Instructional Design (3rd ed.). New York, NY: Holt, Reinhard, & Winston.
- Godfrey, P C, & Grasso, E T. (2000). Working for the common good: Concepts and models for service-learning in management. Paper presented at the American Association for Higher Education.,
- Govekar, M., & Rishi, M. (2007). " Service Learning: Bringing Real-World Education into the B-School Classroom". Journal of Education for Business, 83(1), 3-10.
- Gujarathi, M. & McQuade, R. (2002). " Service-Learning in Business Schools: A Case Study in an Intermediate Accounting Course". Journal of Education for Business, 77(3), 145-150.
- IS2009. (2009). Model Curriculum Draft. from http://blogsandwikis.bentley.edu/iscurriculum/index.php/IS_2009_for_public_review
- Janicki, T., Lenox, T, Logan, R, & Woratscheck, C. (2008). " Information Systems/Technology Employer Needs Survey: Analysis by Curriculum Topic ". Information Systems Education Journal, 6(18).
- Payne, S., Flynn, J. & Whitfield, J. (2008). " Capstone Business Course Assessment: Exploring Student Readiness Perspectives". Journal of Education for Business, 83(3), 141-146.
- Rakes, G. (1996). " Using the Internet as a tool in resource based learning environment". Educational Technology, 6(2), 52-69.
- Rose, J., Rose, A. & Norman, C. (2005). " A Service-Learning Course in Accounting Information Systems". Journal of Information Systems, 19(2), 145-172.
- UNCW. (2009). Mission Statement. from <http://www.uncw.edu/www/7goals.html>
- Woratscheck, C, & Lenox, T. (2002). Information Systems Entry-Level Job Skills: A survey of Employers. Paper presented at the Proceedings of the Information Systems Educators Conference, San Antonio, TX.