AC 2012-4110: THE SOFTWARE AND SYSTEMS ENGINEERING MASTERS PROGRAM AT TEXAS TECH UNIVERSITY: A COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING COLLABORATIVE EFFORT

Dr. Susan Darling Urban, Texas Tech University

Susan D. Urban received the B.S., M.S., and Ph.D. degrees in computer science in 1976, 1980, and 1987, respectively, from the University of Louisiana, Lafayette. She has been a professor in the Department of Industrial Engineering at Texas Tech University since 2011 and was previously a professor in the Department of Computer Science from 2007-2010. She was at Arizona State University from 1989-2007, where she currently holds the status of Emeritus Professor. She was also an Assistant Professor at the University of Miami from 1987-1989. Her research addresses integrated techniques for event, rule, and transaction processing to address data consistency and active behavior in distributed, data-centric applications. Urban has been the recipient of several grants from the National Science Foundation for her research on constraints, active rule processing in centralized and distributed environments, data consistency issues in service-oriented environments, the use of databases in engineering design, undergraduate research, and the development of innovative teaching concepts for database instruction. She has published more than 100 refereed papers and book chapters on the results of her research and is a co-author of Fundamentals of Object Databases: Object-Oriented and Object-Relational Design (Morgan Claypool, 2011). She currently serves in the Editorial Board of the Journal of Data Semantics and has previously served on the Editorial Boards of the IEEE Transactions on Knowledge and Data Engineering and the Journal of Computing and Information Science in Engineering. In addition, she has served as the Co-editor of special issues in the Integrated Computer-Aided Engineering Journal, Computing Systems Journal, IEEE Transactions on Knowledge and Data Engineering, Theory and Practice of Object Systems, and the Journal of Computing and Information Science and Engineering. She has also served on the organizing and program committees of numerous database conferences and workshops, including the IEEE International Conference on Data Engineering, the International Conference on Cooperative Information Systems, the International Conference on Ontologies, Databases, and Applications of Semantics, the International Conference on Objects and Databases, and the ACM Symposium on Computer Science Education. Urban is a member of the Association for Computing Machinery, the IEEE Computer Society, the American Society for Engineering Education, and the Phi Kappa Phi Honor Society. In 2010, she was also inducted into the Golden Key Honour Society as an honorary member.

Prof. Joseph E Urban, Texas Tech University

Joseph E. Urban joined the Department of Industrial Engineering at Texas Tech University as professor in Jan. 2011 after serving as chair of the Department of Computer Science during 2008 to 2010. He served as the Deputy Division Director in the Division of Computer and Network Systems of the Directorate for Computer & Information Science & Engineering at the U.S. National Science Foundation. He has published more than 120 technical papers. He has supervised the development of nine software specification languages. His research areas include software engineering, executable specification languages, prototyping software systems, web based software tools, engineering education, computer languages, data engineering, and distributed computing. He received the Ph.D. degree in computer science from the University of Louisiana at Lafayette.

Dr. Susan Mengel, Texas Tech University

Susan Mengel joined the Department of Computer Science at Texas Tech University in the Fall of 1996 and is currently an Associate Professor. She received her bachelor's degree from Central Oklahoma University, her master's degree from Oklahoma State University, and her Ph.D. from Texas A&M University, all in computer science. While here at Texas Tech, she helped to establish the master's in software engineering degree program, served as the Associate Chair for Graduate Studies, served as Vice-President for the Texas Tech Faculty Senate, chaired the IEEE Software Engineering Education and Training Conference, served on the Steering Committee of the ACM/IEEE Computing Curriculum, and served on the IEEE Computer Society Board of Governors. She currently serves on the Texas Tech Institutional Review Board for the Protection of Human Subjects and is the Associate Editor of Computing for the IEEE Transactions on Education. Mengel conducts research in the area of web search where she is developing techniques to pinpoint needed information more accurately than current search engines, and developing ways to discern if the information on a web page or document is important enough to be archived for later use by a user. In addition, she is investigating the area of information security to determine how to keep intruders out of computer systems and applications, particularly on the web.

Dr. William M. Marcy P.E., Texas Tech University
Dr. Patrick E. Patterson, Texas Tech University

Patrick Patterson is Chair of the Department of Industrial Engineering at Texas Tech University. Previously, while at Iowa State University, he served as Chair of the Industrial and Manufacturing Systems Engineering Department and as Interim chair of the Industrial Education and Technology Department. He is a Professional Engineer (PE), a certified Professional Ergonomist (CPE), and a Fellow of the Institute of Industrial Engineers. His research and teaching interests include safety engineering, interaction design, cognitive ergonomics, user-centered product design, biomechanics, and errors in complex systems. He has extensive experience in developing devices and device adaptations for individuals with disabilities. Recent work includes developing adaptive displays, the effects of product design on human error, evaluating display sophistication on information value, and the design of products for the aging population. He has also developed courses and training packages that use distance learning, streaming video, and interactive distance team collaboration.

The Software and Systems Engineering Masters Program at Texas Tech University: A Computer Science and Industrial Engineering Collaborative Effort

1. Introduction

In several recent reports, software engineering has been identified as one of the best occupations in the job market[1]. Software engineering is a knowledge-intensive occupation, requiring computing professionals with skills that span from requirements elicitation, to software design and implementation, as well as testing, verification, and validation. Software engineers must also have project management and teaming skills coupled with sensitivity to the issues of software and data security. Industry's need for innovation, research and development, and a broader understanding of the complexities of software development is contributing to this growth in software engineering employment opportunities. The Internet and its impact on distributed applications, service-oriented computing, and cloud computing, are also creating a demand for new and better software applications, many involving social computing, ubiquitous and pervasive computing, and mobile computing.

Over the last 20 years, at least 50 graduate software engineering degree programs have been established. The Graduate Software Engineering Reference Curriculum (GSwERC) committee recently conducted a survey of 28 of these programs, finding that many of the programs vary widely in curriculum[2]. GSwERC has also compiled an updated report outlining curriculum guidelines for software engineering programs[3]. An interesting aspect of the report is a suggestion to integrate systems engineering concepts into software engineering degree programs. While software engineers use engineering concepts in the design and development of software systems, computer systems software engineers take a broader view of an organization's computing environment by integrating and coordinating hardware resources, software applications, networking, and security, with an added focus on human factors, workflow, logistics, and technical support. ISO 15288, a systems engineering standard, describes an integrated, software and systems engineering life cycle[4]. As indicated by Dennis Frailey of Raytheon[5], however, poor communication between systems and software engineers is one of the major causes of problems in defense programs. The International Council on Systems Engineering[6] and the National Defense Industrial Association Systems Engineering Division[7] endorse the GSwE2009 curriculum guidelines, which is sponsored by the IEEE Computer Society and Association for Computing Machinery. The complexity of most modern software applications demands a more integrated skill set for systems and software engineers.

To address this need, the Department of Industrial Engineering and the Department of Computer Science at Texas Tech University have collaborated to revise the Master of Science in Software Engineering (MSSE) degree program. The program emphasizes the integration of systems and software engineering concepts. The MSSE is a professional, classroom and/or online, degree program, focusing on developing graduates capable of defining, developing, testing, and maintaining complex software systems by using system requirements engineering techniques that integrate hardware, software, human factors, economic, and application considerations.

This paper presents an overview of the revised MSSE degree program. Background on the GSwERC curriculum is first presented in Section 2. Section 3 then presents the revised systems and software engineering curriculum for the MSSE program, with a comparison to the GSwERC curriculum. Program challenges and future refinements are discussed in Section 4, with a summary in Section 5.

2. Background: graduate software engineering reference curriculum

The first graduate reference curriculum for software engineering was published in the early 1990's by the Software Engineering Institute at Carnegie Mellon University[9], spawning the development of numerous software engineering degree programs, some offering degrees in software engineering and others offering degrees in computer science with a strong emphasis in software engineering. The IEEE Computer Society produced the Software Engineering Body of Knowledge (SWEBOK)[8] in 2004 as a precursor to the development of model curricula and certification programs. The Software Engineering 2004 (SE2004) document[10] was developed by a joint committee of the Association for Computing Machinery and the IEEE Computer Society. This document contains a set of recommendations for an undergraduate software engineering degree program. The core software engineering concepts defined by the SWEBOK include:

- Software Requirements
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Engineering Management
- Software Engineering Process
- Software Engineering Tools and Methods
- Software Quality

The 2004 SWEBOK is currently under revision to better reflect the impact that the Internet has had on software development over the last ten years.

A committee was established in 2007 through Stevens Institute of Technology to develop the GSwERC as a new reference curriculum for graduate software engineering. As part of the development of GSwERC, the committee conducted a survey of 28 software engineering programs[2]. The survey indicates that 25% of the programs are housed in stand-alone software engineering departments, 50% are in computer science departments, and the other 25% are in various other departments. The M.S. in Software Engineering at the University of Texas Austin, for example, is housed in the Department of Electrical and Computer Engineering, while the software engineering program at the Stevens Institute of Technology is housed in the School of Systems and Enterprises. Generally, the faculty size of most software engineering programs is small, with about half of the programs having only five or fewer faculty members. The survey also indicated that 29% of the programs have 25 students or less. About 71% have up to 100 students.

The GSwERC survey[2] also indicates that many programs vary widely with respect to emphasis areas. Some emphasize embedded systems, while other emphasize issues such as software security or defense systems. The most common topics covered are software requirements, software design, and software management. Topics such as software maintenance, software configuration, and software tools and methodology are the least covered topics. Other programs include topics that are not addressed by the SWEBOK, such as networks, human computer interaction, middleware, and information management.

In addition to a revision of graduate software engineering guidelines, the GSwERC addresses the need to integrate software and systems engineering programs. Whereas software engineering uses engineering concepts in the design and development of software systems, computer systems software engineers takes a broader view of an organization's computing environment, integrating and coordinating hardware resources, software applications, and networking and security issues, with an additional focus on human factors, as well as workflow, logistical, and technical support[1]. Most programs still view systems and software engineering as separate disciplines, although the synergy between curriculum groups from each discipline is evident.[4, 6, 8] Faculty, such as those at the University of Texas at El Paso, are building on this synergy through an NSF-supported Science Masters Program, providing a degree program in software engineering that conforms to GSwERC and providing a systems engineering degree program with an emphasis in computer science. The Stevens Institute of Technology also offers several different systems-oriented graduate certificates as a component of its software engineering program.

Table 2 in Section 3 provides an overview of the GSwERC knowledge areas, with the second column of the table highlighting the more systems-oriented content of the GSwERC guidelines. We will first present the curriculum for the MSSE degree program and then discuss Table 2 in more detail through a mapping of the MSSE core courses to the GSwERC knowledge areas.

3. The master of science in software engineering degree program at Texas Tech University

In 1998, the MSSE degree program was approved at Texas Tech University as both an on-campus and a distance program. The initial curriculum was offered in the Fall of 1999 and was inspired by the 1990 Software Engineering Institute graduate curriculum model,[9] which recommended six required courses covering requirements, design, construction and maintenance, verification and validation, project management, and formal methods. The required courses in the MSSE degree program were Software Specification and Design, Software Project Management, Software Process Improvement, and Software Construction and Evolution. A capstone experience was also required through two courses, Software Studio I and II. A set of electives was available and included Software Quality Assurance and Testing, Web-Based Software Systems, and Real-Time and Time-Sharing Systems. The curriculum was later revised to three required courses (Software Project Management, Software Modeling and Architecture, and Software Verification and Validation) and six electives (Special Topics in Software Engineering, Real-Time and Time Sharing Systems, Distributed Systems, Parallel Processors and Processing, Fault Tolerant Computer Systems, and an Industrial Engineering course on Systems Theory).

In the Spring of 2011, a joint effort was initiated between the Department of Industrial Engineering and the Department of Computer Science to revise the MSSE program in the context of the GSwERC guidelines, with specific emphasis on a program that would provide an integrated approach to systems and software engineering. The wide spectrum of application domains that are part of the industry in the State of Texas needs to further address the integration of systems and software engineering education for more effective development of systems. This leadership role can be replicated outside the State through the offering of online degree programs.

Faculty from both departments met to discuss the relevant issues, with a goal of having a program in place by the Fall of 2011. When bringing faculty together from different departments, there are always cultural differences between the different disciplines and thus a curve associated with learning how to truly develop an integrated approach to teaching an interdisciplinary topic. Our approach has been to make the program available with current resources and to implement a continuous review and revision process as we learn more from each other about teaching systems and software engineering topics in a more integrated way.

Table 1 presents the curriculum for the revised MSSE program. Applicants should have a B.S. degree in a computing-related discipline with proficiency in probability and statistics *or* a B.S. degree in an engineering discipline with proficiency in at least one high level programming language. The MSSE is a 30 credit hour program. Students are required to take four core courses, two of which are from computer science (Software Modeling and Architecture, and Software Verification and Validation) and two of which are from Industrial Engineering (Project Management and Systems Theory). We chose to provide the broader systems engineering perspective of project management as part of the core, with software project management available as an elective. Systems theory was an elective in the previous program and has been elevated to a core course.

Five elective courses are then selected from computer science and/or industrial engineering options. As indicated in Table 1, the computer science (CS) electives are focused on software project management, web-based software systems, fault tolerant computing, and advanced data management. A special topics course is also typically offered every semester, with topics such as software reliability engineering and software security. The industrial engineering (IE) electives emphasize usability engineering, control theory, Bayesian analysis, risk assessment, risk modeling and assessment, and economics of systems. Similar to computer science, a special topics course is also available. Ethics in engineering is also included to emphasize the importance of ethics in the engineering of complex software systems.
The curriculum includes a 3-credit hour capstone design and implementation project. The project is a group project requiring that students work in teams to address the systems and software engineering aspects of the project.

Table 1: MSSE degree program with systems and software engineering focus

CORE COURSES (12 Hours)
CS 5373 Software Modeling and Architecture. This course introduces the theory and practice for software development and covers software requirements, analysis, software architecture and detailed design.
CS 5374 Software Verification and Validation. This course introduces how to implement effective test and measurement programs as well as how to apply this knowledge to the production of low-defect software.
IE 5329 Project Management. This course covers technical, organizational, and personnel project management examination including planning, estimating, budgeting, scheduling, resources management, and control. It also includes risk analysis and management using software for project performance evaluation.
IE 5320 Systems Theory. This course examines theoretical foundations of general systems theory applied to engineering and organizational enterprises addressing issues of systems efficiency, effectiveness, productivity, economics, innovation, quality and QWL.
Electives (15 Hours)
Computer Science Electives
CS 5363 Software Project Management. Explores the principles of software project management and their effective application. Topics include project, risk, process, and resource management and improvement techniques.
CS 5369 Web-Based Software Systems. In-depth study of how to engineer Web-based software systems. Topics include process, development, testing, and performance issues.
CS 5380 Fault Tolerant Computing Systems. Introductory course to methodologies for specifying, designing, and modeling fault-tolerant computer systems. Includes fault classification, design techniques for fault detection and recovery, and reliability modeling techniques.
CS 5356 Advanced Database Concepts. Systems aspects of relational databases are emphasized. Topics include relational database design, index and access structures implementation and performance evaluation, query processing and optimization, transaction management, and concurrency control.
CS 5332 Special Topics in Software Engineering. Studies in Advanced Software Engineering. May be repeated for credit. Past topics include software reliability engineering and software security.
Industrial Engineering Electives
IE 5301 Usability Engineering. Usability fundamentals, measuring usability, the usability engineering lifecycle, design techniques, heuristics for improving usability, user testing, assessing usability, interface standards, and internationalization.

Table 1 (continued)

IE 6304 Control Theory. Cybernetics; feedback and feed-forward; Fitts' law; linear systems; Laplace transforms; gain and lag; Fourier analysis; coherence; stochastic resonance; frequency domain; bode analysis; optimal control law.
IE 5302 Bayesian Analysis. Subjective probability; satisficing; Hurwicz principle; signal detection; ROC linearization; cross-entropy; Kullback–Leibler divergence; discriminant analysis; Monte hall; Bayesian net; data envelopment.
IE 5308 Risk Assessment. Risk perception; psychophysics; multinomial logic choice; competing risks; life regression; proportional hazards; multi-objective and multi-attribute decision models; group decisions; Choquet integral; copula; social networks.
IE 5319 Risk Modeling and Assessment. Probabilistic risk models; probability distributions for risk modeling; input data for risk modeling; low probability events; risk modeling software; and analysis of risk modeling results.
IE 5324 Advanced Economics of Systems. This course studies sensitivity of engineering economics factors and Monte Carlo approaches to sensitivity analysis. It also studies economic performance measures, including analysis and modeling for automated manufacturing systems.
IE 5332 Experimental Investigation in Advanced Industrial Engineering Applications. This course provides students with the opportunity for individual experimental study of advanced topics selected on the basis of departmental recommendation. May be repeated for credit.
ENGR 5392 Ethics in Engineering Practice and Research. Applications of professional ethics to engineering practice and research in fields of education and technology-related industry.
FINAL PROJECT (3 Hours)
CS 5358 Software Studio I. Capstone design and implementation of a project involving the integration of system and software engineering techniques.

The core courses were determined by mapping several CS and IE courses to the GSwERC Core Body of Knowledge (CBOK) and selecting those that provided a reasonable foundation in the GSwERC knowledge areas, leaving flexibility to cover other knowledge areas and related topics through the program electives. For example, CS 5363 Software Project Management and IE 5329 Project Management are related in that they both address project management concepts, such as work breakdown structures, scheduling, risk management, client relationship, and team management. Software project management addresses management techniques specifically related to the production of software, such as software process management, software process improvement, the capability maturity model, configuration management, software estimation models such as COCOMO, and software deployment. The project management course that is part of the core takes a broader view of project management issues, such as project management organization, contract negotiation, resource management, and external processes such as manufacturing.

The GSwERC CBOK is shown in column 1 of Table 2. The second column of the table indicates the topics that represent the systems-oriented knowledge areas as defined in the GSwERC guidelines[3], which include ethical issues, fundamentals of systems engineering, requirements engineering, testing, maintenance, risk management, engineering economics, and quality. The remaining columns of Table 2 show the four core courses of the MSSE program. As indicated, the CS and IE core courses cover the knowledge areas of ethics and professional conduct, systems engineering, requirements engineering, software design, software testing, software engineering management, software engineering process, and software quality. CS 5373 Software Modeling and Architecture covers many knowledge areas, so the instructor must judiciously decide which areas to cover more deeply than others. The course brings depth of coverage to areas C (requirements engineering) and D (software design) through consideration of architectural and design patterns, problem domain modeling, function lists, UML, and formal modeling. The other knowledge areas are addressed in relationship to the requirements and design material.

The areas covered by the core also subsume most of the systems engineering content of the GSwERC guidelines. The areas not covered by the core courses are software construction, software maintenance, and configuration management. Offering such courses on a regular basis is a matter of faculty expertise and other workload issues. These topics can currently be covered through varying subjects offered in CS 5332 and IE 5332.

We also mapped the core courses to the INCOSE Systems Engineering Body of Knowledge (SEBOK)[6]. As shown in Table 3, the core courses of the program also provide coverage of relevant systems engineering competencies related to business processes, system architectures, life cycle cost and benefit analysis, and modeling simulation and analysis issues.

4. Program refinements and challenges

We have identified four areas of focus for strengthening the program:

Additional software engineering courses and electives. In addition to the knowledge areas not covered by the core courses, specific software engineering additions under consideration are requirements engineering, software evolution, and service-oriented computing. We are also evaluating additional electives, such as networking, distributed computing, and parallel processing, especially considering the role of distributed and parallel computing in the development of software systems.

Software Security Track. We have identified security as an area for defining a track within the MSSE program. A security track will support current research within the university related to security for the smart grid. Combined with some of the risk management electives from the IE department, this will give students the option to acquire a certificate in software security as part of the MSSE degree program. Specific courses of interest include software security, data and information security, software reliability, software quality assurance, network security, and cyber security. Other programs should be able to adopt our program model with or without the security track.

Table 2: Mapping of MSSE core courses to the GSwERC knowledge areas

GSwER2009 CBOK Knowledge Areas	Sys. Eng. Content	CS 5373	CS 5374	IE 5329	IE 5320
A. Ethics and Professional Conduct					
1. Social, Legal, and Historical Issues	SYS			X	X
2. Code of Ethics/Professional Conduct	SYS			X	
3. Role of Software Eng. (SwE) Standards		X			
B. Systems Engineering (SE)	SYS				
1. SE Concepts				X	X
2. SE Life Cycle Management				X	X
3. Requirements				X	
4. System Design					X
5. Integration and Verification			X		
6. Transition and Validation			X		
7. Operation, Maintenance, Support					X
C. Requirements Engineering (RE)	SYS				
1.Fundamentals of RE		X			
2. RE Process		X			
3. Initiation and Scope Definition		X			
4. Requirements Elicitation		X			
5. Requirements Analysis		X			
6. Requirements Specification		X			
7. Requirements Validation		X	X		
8. Practical Consideration		X			
D. Software Design (SD)					
1. SD Fundamentals		X			
2. Key Issues in SD		X			
3. Software Structure and Architecture		X			
4. SD Quality Analysis and Evaluation		X	X		
5. SD Notations		X			
6. SD Strategies and Methods		X			
E. Software Construction (SC)					
1. SC Fundamentals					
2. Managing Construction					
3. Practical Considerations					
F. Testing	SYS		X		
1. Testing Fundamentals			X		X
2. Test Levels			X		
3. Testing Techniques			X		
4. Test-Related Measures			X		
5. Test Process			X		
G. Software Maintenance (SM)	SYS				
1. SM Fundamentals					
2. Key Issues in SM					
3. Maintenance Process					
4. Techniques for Maintenance					

Table 2 (continued)

H. Configuration Management (CM)					
1. Management of the CM Process					
2. Configuration Identification					
3. Configuration Control					
4. Configuration Status Accounting					
5. Software Release Mgmt and Delivery					
I. Software Engineering Management					
1. Software Project Planning				X	
2. Risk Management	SYS			X	X
3. Software Project Org. and Enactment				X	X
4. Review and Evaluation		X		X	X
5. Closure	SYS			X	
6. Software Engineering Measurement		X		X	X
7. Engineering Economics	SYS			X	
J. Software Engineering Process					
1. Process Implementation and Change					X
2. Process Definition					
3. Process Assessment					X
4. Product and Process Measurement	SYS				X
K. Software Quality (SQ)			X		
1. SQ Fundamentals	SYS		X		
2. SQ Management Processes			X		X
3. Verification and Validation	SYS	X	X	X	

Integrated CS and IE courses. Our most important challenge is to identify how to truly create an integrated offering of software and systems engineering courses. This process will involve more interaction between CS and IE faculty about how to weave together disciplinary concepts into courses that are possibly even team-taught by faculty from each discipline. For example, we would like to define a systems and software engineering course as a core course. Project management, requirements engineering, software reliability, software quality assurance, and intelligent manufacturing are also topics that can be taught from a more integrated perspective. Courses in the planned security track also need a software and systems perspective, especially if we build on the smart grid research program.

Integrated software and systems engineering final projects with industry involvement. Our current plan for the final project course is for students to work in teams to address problems that require an integrated approach to systems issues and software issues. We anticipate that, as a result of the program, students within each team will have strength in one of these areas, with an appreciation and understanding for the other area. To develop truly meaningful projects, we will eventually seek partnerships with industry for definition, administration, and evaluation of project results.

Table 3: Mapping of MSSE core courses to the INCOSE systems engineering competencies

SEBOK (SE Competencies from INCOSE)	CS 5373	CS 5374	IE 5329	IE 5320
1. Business Processes/Operational Assess.				
a. Business & Operational Shortfalls			X	X
b. Solution Requirements				X
2. System/Solution/Test Architectures				
a. Develop functional architecture				X
b. Develop physical architecture				
c. Allocate functions to physical elements				X
d. Develop interface architecture				
e. Develop test system architecture		X		
f. Determine/manage impact to solutions				
g. Develop manufacturing system arch.				
h. Develop the training system arch.				
i. Develop the deployment system arch.				
3. Life Cycle Cost & Cost Benefit Analysis				
a. Conduct life cycle costing				X
b. Perform activity-based costing				
c. Cost benefit analyses of SE process			X	
d. Understand the system cost drivers				X
e. Estimate total ownership cost				X
4. Serviceability / Logistics				
a. Develop system maintenance concept				
b. Develop system supply support				
c. Analyze operational/servicing skill req.				
d. Develop system and platform docs.				
5. Modeling, Simulation, & Analysis				
a. Performance modeling/forecasting		X		X
b. Architecture modeling/analysis				
c. Reliability/availability/maintainability		X		X
d. System safety analysis		X		X

5. Summary and conclusions

This paper has provided an overview of the integrated systems and software engineering MSSE degree program at Texas Tech University. The program became effective in the Fall of 2011 and we are just in the beginning stages of program assessment and revision. There are currently 16 students enrolled in the program, with five students on campus and 11 distance students. Several of these students are still carry-overs from the previous curriculum, although numerous students have been admitted under the new curriculum over the Summer-Fall 2011 timeframe. Our goal is to at least double enrollment over the next five years through advertisement of the program within the defense industry as well as energy, communication, and manufacturing industries that have an understanding of the interdisciplinary skill set that is offered through the program. We expect the curriculum to evolve over the next few years as faculty from each discipline gain a better understanding of the pedagogical aspects of interdisciplinary instruction and as we recruit greater participation from industry in review of the curriculum and participation in course projects. We also see the MSSE degree as a pathway to the online Systems and Engineering Management Ph.D. program offered by the Department of Industrial Engineering, which provides a computing area of emphasis.

Acknowledgments

We would like to thank the faculty from the Department of Computer Science and the Department of Industrial Engineering that contributed to the revision of the integrated MSSE degree program. In particular, we would like to thank James Smith, Jennifer Farris, Akbar Siami-Namin, Barbara Millet, Simon, Hsiang, and Sunho Lim.

References

1. Bureau of Labor and Statistics, *Occupational Outlook Handbook*, 2010-2011 Edition, http://www.bls.gov/oco/ocos303.htm.
2. Pyster, A., Turner, R., Henry, D., Lasfer, K., and Bernstein, L., "Master's Degrees in Software Engineering: An Analysis of 28 University Programs," *IEEE Software*, September/October 2009, pp. 94-101.
3. *Graduate Software Engineering 2009: Curriculum Guidelines for Graduate Degree Programs in Software Engineering*, Integrated Software and Systems Engineering Curriculum Project, Stevens Institute of Technology, 2009, http://www.gswe2009.org/.
4. ISO 15288:2008 *Systems and Software Engineering – System Life Cycle Processes*, International Organization for Standardization, 2008.
5. Frailey, D., Educating the Next Generation of Software & Systems Engineers, Ratheon Software Council Presentation, May, 2009.
6. International Council on Systems Engineering, http://www.incose.org, 2012.
7. National Defense Industrial Association Systems Engineering Division, http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Pages/default.aspx, 2012.
8. Abran, A., Moore, J., Bourque, P., and Dupuis, R., Guide to Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society, http://www.computer.org/portal/web/swebok, 2012.
9. Ardis, M.A., and Ford, G.A., *1989 SEI Report on Graduate Software Engineering Education*, Technical Report CMU/SEI-89-TR-2, Carnegie Mellon University, Pittsburgh, PA, 1989.