Design and Assessment of a Multidisciplinary Course in Service Oriented Architecture

Thomas Way and Vijay Gehlot
Center of Excellence in Enterprise Technology
Department of Computing Sciences
Villanova University, Villanova PA 19085
thomas.way@villanova.edu
vijay.gehlot@villanova.edu

Abstract - *Service Oriented Architecture (SOA) is finding increasing acceptance in industry for the design of enterprise-scale, networked applications. However, opportunities to study SOA in appropriately interdisciplinary ways are currently lacking for college students. In this paper, we present the design of a graduate and upper division undergraduate level course that combines elements of business, engineering and computer science in a real-world-inspired approach. The design of the course is presented and motivated, objectives and expected outcomes defined, and assessment data from the initial offering of the course are given. An analysis of the elements of the course is provided, resulting in guidelines for the design of a relevant SOA course. Based on these results, a modified plan for a second offering is described.*

Keywords: Service oriented architecture, computer science education, preparing graduates for industry.

1 Introduction

Service Oriented Architectures (SOAs), the techniques for developing and deploying loosely coupled, interoperable, implementation independent network services that exchange data with each other [2], are continuing a trend toward increased adoption by industry [1]. The technology is sufficiently mature, and the available software tools make it possible to develop SOA services rapidly. Although the underlying design, development and networking issues are often complex, well-trained software engineers can master SOA and most of the technical challenges are quite solvable. Thus, the key issue preventing broader adoption of SOA is the practice of governance [6], which includes establishing agreed upon standards for interoperability and the significant challenge of balancing the competition of the marketplace with the inherent cooperation and collaboration required for SOA to be successful.

The current trend in SOA is the "enterprise mashup," a large-scale, revenue-creating web application which, like a regular mashup, integrates data from multiple sources to produce a merged online product or tool [5]. For example, an enterprise mashup could be used by a real estate web site to correlate data about neighborhoods with house sales information for a specific real estate agent. The business model is to charge the agent only for the targeted data being used, similar to an Apple iTunes model where songs rather than entire albums can be purchased [8]. SOA of this sort will continue to require significant governance efforts if it is to be successful. Figure 1 illustrates how enterprise-scale mashups consume and aggregate data from multiple producers, and how commercial and non-commercial data providers may interact.

It is into this marketplace of significant but solvable software engineering challenges and identifiable but potentially intractable business practice conundrums that our graduates enter. The best prepared newcomers to the real-world application of SOA should be astute on the technical issues of enterprise-scale, networked, software design and deployment and also on the motivating business aspects that are manifest in the natural tension between competition and collaboration. Computer science graduates who plan to use SOA in their careers need to augment their learning with a broader view of the topic. The computer

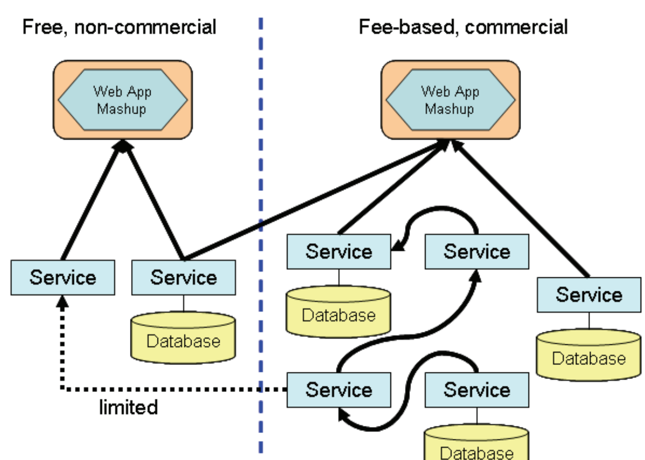

Figure 1. Enterprise Mashups, an emerging application of SOA technology. Diagram shows producer-consumer relationships among commercial and non-commercial services.

scientist cannot focus purely on the technical aspects of SOA, but must address the business and people side of things more than ever.

The rapid adoption of SOA means that education of software engineers is frequently performed on-the-job, rather than in a formal classroom setting. Software engineers have retrained themselves as needed to incorporate this emerging technology in their businesses. The learning curve can be significant, and tight development schedules mean the focus is often on technical issues. The result is that governance aspects that are becoming an inextricable component of the design process can be initially overlooked. Opportunities for education are few, but growing. In addition to a small number of vendor-offered, industry-oriented training seminars, when the proposal to develop this course was first made only Brandeis University offered a course specifically on Service Oriented Architecture. In a recent online search, only six universities were found that offered a graduate or undergraduate course specifically on Service Oriented Architecture, namely Brandeis University, Heinz College, Indiana University, Stevenson University, University of Colorado, and Virginia Community College. Most of these, however, appear to focus almost exclusively on the implementation and programming aspects of SOA. We anticipate that many other universities will develop SOA course offerings as the demand for SOA training and education increases.

There are precedents within our discipline for use of a multidisciplinary approach to computer science courses, with the most frequent example being those in the area of software engineering [9]. Software engineering curricula includes technical topics of software tools, methods and design, and typically acknowledges the business aspects needed to manage a software project and to successfully bring that project to market. Topics covered in a software engineering course include software design and project management, software specification, implementation approaches and development tools. Courses also include topics such as risk assessment, product lifecycle, and cost estimation from the business world, and process models and project management from operations management or engineering disciplines.

The incorporation of sufficiently complex, real-world projects [4] and exposure to industry practices and experts [11] into a software engineering course provide students with valuable experience applying technical concepts and learning software development from an industry perspective. Where most software engineering courses fall short where SOA is concerned, however, is in broadening the notion of business methods to include the need for collaboration among competitors. This is the key innovation of, and difference from past methods, that SOA brings to the subject of software engineering. The current dearth of formal SOA coursework, coupled with the expertise developed by faculty researchers at our institution gained through a DoD-funded, enterprise-architecture research project, motivated the development of a multidisciplinary course for graduate and upper-level undergraduate computer science students.

2 Course Design

The designed course in Service Oriented Architecture incorporates three major elements, three primary learning objectives fulfilled by a number of outcomes, a broadly varied selection of content to support the objectives, and an assessment plan to provide evidence of successful approaches, student learning and course design flaws. The semester course met one evening per week for 2¾ hours, with an expectation that significant student work will be performed outside of class meetings.

2.1 General Elements

The course is comprised of three primary elements with the intent of providing coverage of computer science, engineering and business aspects of SOA. First, an understanding of the issues and technologies from computer science, engineering and business is gained through lecture, guest speakers, readings, and written assignments. This approach is important because it provides a deep understanding of the foundational issues related to SOA and its successful adoption and deployment.

Second, hands-on application of these concepts is explored through a major, collaborative project in which students work in small teams to produce SOA services that interact with services produced by other students. This collaborative project mimics the real-world situation of companies, or other information providers, partnering to provide needed information to each other while maintaining the ability to compete against each other in the marketplace. This project should be drawn from real-world examples, such as a distributed airline scheduling and reservation system, national weather information network system, or large-scale health care information system.

Third, modeling and simulation of SOA at an enterprise-scale are performed, using industry-standard tools such as OPNET and CPN Tools. As the complexity and scope of networking has increased, so too have the monetary stakes involved in deploying large, or enterprise, scale systems. For example, the major problems that US Airways encountered when it merged two enterprise-scale information systems in March, 2007 [10] highlighted the need for an understanding not merely of the computer science of enterprise networking such as SOA, but of related business and engineering issues. Students will learn through experimentation that modeling and simulation can test expensive systems in a “sandbox” to identify and address design flaws before implementing them for real, thereby reducing business and engineering risk.

The topics covered during the semester were organized into a number of categories:

- **SOA Concepts**: service provider, service consumer, service discovery
- **SOA Benefits**: interoperability, agility, cost
- **SOA Challenges**: dynamic discovery, security, uniform standards, performance, service guarantees, governance, service composition, service granularity
- **SOA Technology/Approaches**: BPEL, ESB, WSDL, UDDI, XML, Web Services, REST, HTTP
- **SOA Modeling**: CPN, OPNET

2.2 Learning Objectives and Outcomes

Three learning objectives were developed to provide students with the technical and theoretical foundations of SOA with context from computer science, engineering and business disciplines. Each objective is accomplished by fulfilling two or three outcomes.

Objective 1: Survey the topic of Service Oriented Architecture (SOA), including related aspects of computer science, engineering and business, and how these relate within the larger topic of software engineering.

- **Outcome 1.1**: Student will demonstrate an understanding of the topic of SOA within the larger context of software engineering, including service providers, service brokers, service requestors, and the approaches, goals, benefits and methodologies for successfully implementing, adopting and deploying SOA.
- **Outcome 1.2**: Student will demonstrate an understanding of the business and technical challenges inherent in SOA, including issues of cooperation, competition, governance and interoperability.

This first learning objective and corresponding outcomes were accomplished with lectures, guest speakers, assigned readings, discussions and student presentations that covered a range of subject matter.

Objective 2: Provide experience developing SOA applications either individually or as part of a team.

- **Outcome 2.1**: Student will be able to implement a SOA application, including a service provider and service requestor.
- **Outcome 2.2**: Student will demonstrate an understanding of the business and technical challenges inherent in SOA by addressing issues of competition and cooperation by designing an interoperable SOA application as part of an individual or team design project.
- **Outcome 2.3**: Student will successfully apply SOA design principles and practices by implementing an interoperable SOA application as part of an individual or team design project.

The second objective and its outcomes were accomplished with a team-based project, and numerous related lectures, laboratory exercises and reading assignments. The large-scale, final project served as a focal point of learning for the course, linking together the variety of material from the computer science, engineering and business disciplines into a meaningful whole. In addition to reinforcing the technical aspects of SOA, it gave students first-hand experience with the business and people aspects involved with managing a team working toward a common goal and of encouraging collaboration among competitors, in this case among other teams in the class.

Objective 3: Learn about modeling and simulation of Service Oriented Architecture, and gain experience using current modeling and simulation software.

- **Outcome 3.1**: Student will be able to create a model and simulation of a SOA application using one or more current academic or professional modeling and simulation software tools.
- **Outcome 3.2**: Student will demonstrate an understanding of the business and technical challenges inherent in SOA by addressing issues of communication, service brokering, service providing, service requesting, and other technical issues of SOA by creating a model and simulation of a SOA application.

The third objective and outcomes where accomplished using the CPN Tools and OPNET modeling software to provide hands-on application of SOA concepts through a variety of laboratory exercises as individuals and in teams.

2.3 Content Development

Lacking a single, comprehensive SOA textbook and little existing classroom tested course material presenting the subject from computer science, engineering and business perspectives, a significant quantity and variety of course content was developed.

Two **textbooks** on Service Oriented Architecture were selected to provide students with a good foundation, broad context and practical implementation techniques on the subject, although the texts were augmented heavily with readings from online sources. The texts used were *Service-Oriented Architecture (SOA): Concepts, Technology, and Design* by Thomas Erl [2] and *SOA: Using Java Web Services* by Mark D. Hansen [3].

Selected readings from online sources were assigned to support the lecture topics. Online sources were used because of the rapid changes in the SOA field, to be as current as possible, to provide a broad range of subject matter that no single text currently provides, and to offer experience with the same materials that software professionals working in SOA use every day.

Homework assignments were given as part of the selected readings from online sources, from the textbook and as smaller scale programming or modeling projects as a way to motivate and assess student learning throughout the semester.

Lectures using PowerPoint slides were presented for one-quarter to one-third of each class meeting to introduce and reinforce material from assigned readings, and were made available to students ahead of time to assist with note-taking. The topics, in the order presented during the semester, were: Introduction to SOA, XML, Web services in Java, WSDL, BPEL, IBM Rational Tester for SOA Quality, JMeter and Performance Testing, ESBs, JBoss, SOAP, REST, UDDI, XML Compression, Binary XML, Performance Testing, SOA Modeling, QoS Issues, Governance, and SOA Security. These topics were reinforced, and many other topics were covered briefly, as part of other activities of the course.

Guest speakers presented first-hand accounts of their professional experiences working with technical, engineering and business aspects of SOA in industry, offering talks on Service Component Architecture (SCA) and a provocative buzzword-driven analysis of enterprise services entitled “2010: An Acronym Odyssey.” Identifying, contacting and scheduling guest speakers with appropriate experience was the most challenging aspect of content development. Guest speakers were located by contacting alumni and colleagues in academia and industry and asking for suggestions and introductions. Although a small stipend was offered, scheduling was much more of an issue than cost, with a number of potential guest speakers unavailable to speak during the weekly class meeting time. Email and telephone correspondence were used to make all arrangements, which were quite time-consuming.

Laboratory activities enabled students to learn the practical aspects of SOA software development using industry standard tools, with a focus on NetBeans. In other activities, students learned to model enterprise scale architectures using mathematical modeling with Colored Petri Nets using CPN Tools and to model network architectures and measure their performance with the professional network modeling software package OPNET.

Topic presentations were assigned, with each student selecting an advanced topic from any relevant aspect of SOA and preparing and delivering a talk to the class on that topic. This gave students experience with independent study of a topic to greater depth, and gave all students exposure to a broad range of advanced topics as they listened to the presentations.

Team-based projects offered students the opportunity to develop and practice real-world collaborative skills in the development of models and SOA services. Team-based projects provided students with the technical skills and experience necessary to work together on the final project.

To conclude the semester, a **large-scale final project** involved student teams creating interoperable services as part of a “smart house” simulation, with services providing “live” data on internal home environment data sources such as temperature, energy consumption, and food inventory, and external data sources such as banking balance, grocery ordering, and power use billing. This project gave students first-hand experience with many of the non-technical issues related to competition and cooperation that are critical to the successful deployment of SOA services in the marketplace. Design of the final project was the most labor intensive aspect of content development, as examples and a relatively complex working solution needed to be developed to test the viability of the project idea prior to assigning the task to students.

3 Assessment

To assess student learning and measure the effectiveness of the course design, students were provided with an initial and concluding survey, project assignments, and an exam. The survey was given to 13 students, and in its initial form gathered some basic demographic information. Of the 13 students, 11 had earned a BS degree in computer science, with one BS each in business administration and finance. Software engineering experience was categorized into “over 5 years industry experience,” “3 to 5 years industry experience”, “0 to 2 years industry experience” and “only as part of coursework.” The range of student experience was distributed relatively evenly into these categories, with a couple more students falling into the “0 to 2 years” category than into the others.

The majority of students indicated that they had some professional business management experience with 3 having served as managers and 8 (include the 3 managers) having performed project leadership duties. The remaining 5 students reported no previous business experience.

Students were asked about their programming experience and which programming languages they were fluent in, with the breakdown shown in Figure 2.

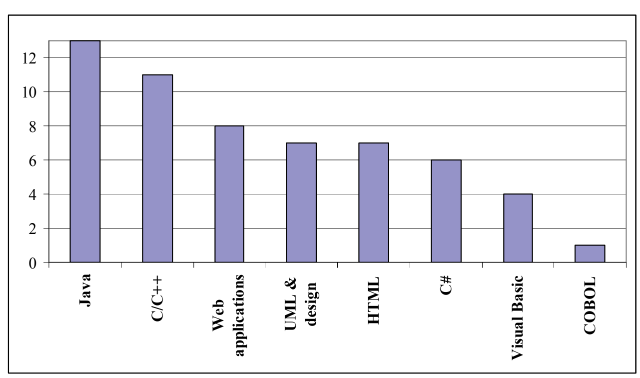

Figure 2. Distribution of student programming language fluency at start of semester.

In order to measure learning, students were surveyed at the start and finish of the semester regarding their understanding of SOA with regards to software design (Figure 3), business issues (Figure 4), engineering, networking and hardware issues (Figure 5), and general terminology or SOA literacy (Figure 6). Students reported that the level of their understanding from choices: excellent, very good, good, average, fair and poor. As illustrated in the following four figures, all students reported improved understanding in all categories measured.

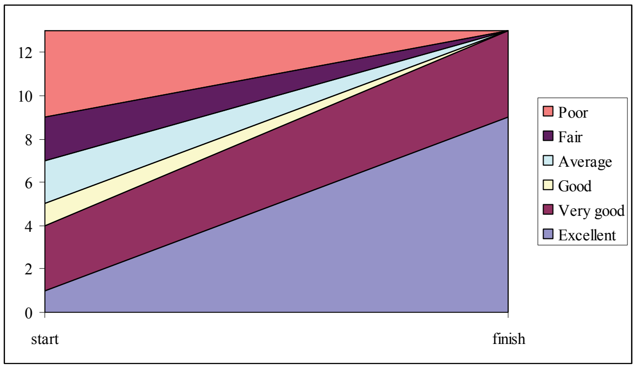

Figure 3. Student self-assessment of understanding of SOA software design at start and finish of semester.

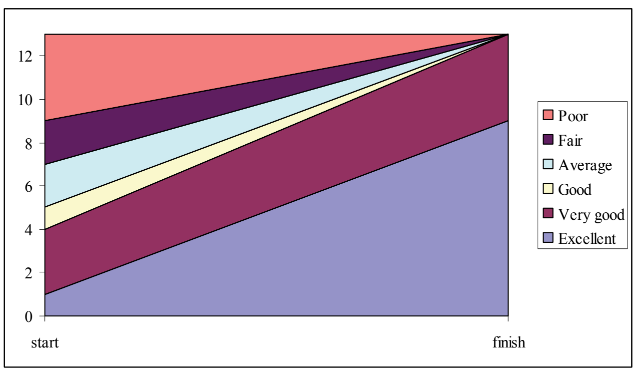

Figure 4. Student self-assessment of understanding of business issues of SOA, including governance, adoption, cooperation and competition at start and finish of semester.

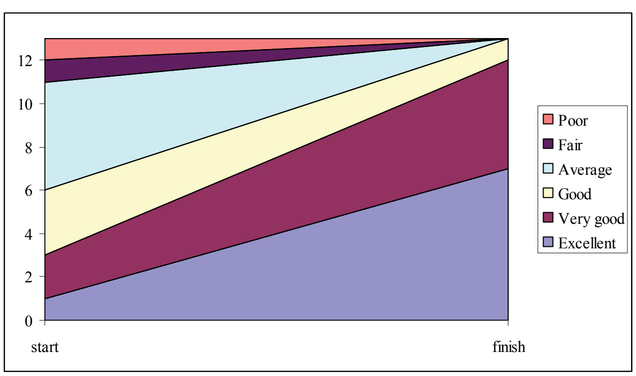

Figure 5. Student self-assessment of understanding of SOA-related engineering, networking and hardware issues at start and finish of semester.

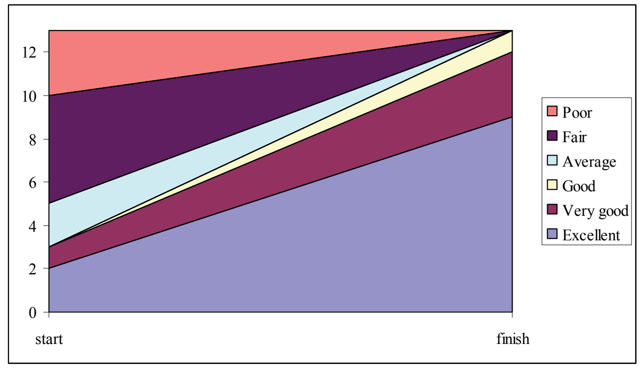

Figure 6. Student self-assessment of familiarity with SOA terminology and general SOA literacy at start and finish of semester.

As part of the semester-concluding survey, students were asked for feedback on various aspects of course content, specifically: topic coverage, reading material, guest lectures, exposure to current tools and technologies and SOA programming exercises. All 13 students reported that topic coverage was adequate, with numerous comments about liking the wide variety of topics. One student specifically suggested an increased emphasis on SOA business process management would be beneficial to students entering the workforce.

Reading material was judged adequate by 11 students, with 2 indicating that they felt the material was inadequate, although for an unexpected reason. Among those who found the readings adequate, a number commented that there was no need for a textbook because of the wide availability of material online. The 2 inadequate ratings for the reading material were given as a result of the material being too advanced for students with less technical backgrounds, although the text was still judged useful because it was supplemented with hands-on learning and discussion that aided understanding of the more technical material.

Two guest lecturers spoke during the semester, and 10 students found these guest lectures to be adequate, serving as a “good break” from the other course activities. The 3 students who found the guest lectures to be inadequate commented on the repetition of some topics, the overly advanced nature of other topics, and a desire to hear more anecdotes about the actual challenges, successes and failures faced by industry professionals.

Current software tools and SOA technologies were covered in class, and students reported on this exposure with 12 stating the coverage was adequate and 1 student feeling it was inadequate. Students felt they gained a broad appreciation of the technology, security and business issues, and found the projects to be the most valuable activity for improving understanding. The student who found that the coverage of SOA tools and technologies was inadequate did so because of the steep learning curve involved and a desire to have more laboratory time to master the tools.

The use of SOA programming exercises as part of laboratory exercises and small scale programming projects was deemed adequate by 8 students and inadequate by the other 5 students. Learning-by-example was the common reason given for liking the exercises, while there was a desire for more coverage of SOA programming in lectures and the feeling that more was learned from lectures and readings than from completing programming projects.

Students were also asked to provide suggestions for how to improve the course, and those suggestions included:

- More programming (many said this).
- More focus on the course project, more time in-class to work on it.
- Final team project was too large, smaller groups would be better.
- Intermediate (midterm) project to reduce size of culminating final project.
- Have a couple easier projects at first before the larger and more difficult one.
- Better organization of the project.
- Difficult to coordinate large group, maybe have smaller groups in future.
- Don't have student topic presentations because we learn more from professor.
- More in-class programming activities or labs.
- More formal structure for class overall.
- Great as graduate course, but needs more structure for undergraduate course which would be good for resumes.

Additional evaluation was performed on an ad-hoc basis throughout the semester, through student interactions, reactions to projects, and ongoing subjective assessment of each activity and assignment. These observations were used to adapt activities and content during the semester as needed, particularly when it was clear that students needed more time to absorb concepts and make use of them in projects. In addition to course specific evaluation, a university-mandated, standardized course evaluation survey was performed near the end of the semester, with data being made available a few months later.

4 Analysis and Adaptations

Overall, student satisfaction with the course and its content was very high, with a few noted exceptions. Students found the reading assignments from the textbook and online sources to be valuable, but some thought that readings were too advanced at times and textbooks were not necessary due to the preponderance of free and up-to-date reading material online. As a result of this feedback, we re-evaluated the available textbooks and selected a new text for the second offering of the course. The text that was selected is *Applied SOA: Service-Oriented Architecture and Design Strategies* by Michael Rosen, et al. [7], which was chosen for its accessible presentation of material, good overview and design-oriented approach.

Traditional lectures are the staple of most courses, and students found the lectures provided to be of just the right quantity and quality. Some expressed the desire for more lecture or discussion devoted to learning to use the various SOA technologies. To accommodate this need in the future, we plan to more diligently assess the technical backgrounds of each student. If many students lack a strong technical background, more time will be spent introducing SOA design and programming tools and techniques. In the event that only a few students need extra help in this regard, individualized tutorials will be assigned to assist these students who come in with less technical backgrounds to get up to speed more quickly.

Homework and laboratory assignments were well received, involving a combination of selected readings and outside-of-class laboratory exercises. In order to better prepare all students for the larger project later in the semester, hands-on laboratory activities will be assigned in the next version of the course to get students working with the technology earlier.

Students found guest speakers to be informative, but there was a desire to minimize repetition of material, limit what was perceived as overly-technical content, and provide more discussion of practical SOA design and development experience from industry. Locating and scheduling guest speakers was a very time-consuming process, so in the next offering of the course we will begin much earlier to identify and prepare for guest speakers who can meet the expressed needs of students.

Programming projects, in particular the large-scale final project, were viewed by students as invaluable to their learning. Students expressed a desire for more and earlier programming projects as a way to spread out the work load and make the learning curve less steep. As a result of the significant amount of feedback regarding project content, the next version of the course will incorporate hands-on programming, tool use and modeling projects earlier in the semester.

Students expressed a desire for a more defined organization to the course, a desire expressed by the course instructor as well. Due to the ambitious agenda and broad scope of the course, and to its newness, there are many opportunities to improve the course organization in future offerings. The next offering will benefit from the initial one, with valuable feedback from students already helping to reform the course into a much improved edition. One other way to add additional organization and structure to the course is that the original, faculty-maintained course web page will be transferred to our institution's virtual learning environment system (WebCT / Blackboard). In this way, some of the considerable time spent in the first offering on designing, organizing and maintaining online content for the course will be reduced, enabling time to be better spent on course management, content preparation and overall organization.

The results of the institutionally mandated Course and Teacher Surveys (CATS) surveys were analyzed, and we were encouraged that our internal course surveys were supported by these results. Students were generally pleased with the course and indicated that an appropriate degree of challenge, variety and content was provided. Indications for greater overall course organization and earlier and more frequent opportunities to perform hands-on projects and laboratory exercises also matched the feedback we previously received.

Student grades for the course were good to excellent, with most of the students exhibiting high motivation, timely completion of work, and the desire and ability to produce high quality work. Not surprisingly, students with stronger technical backgrounds coming into the semester performed better on the more technical aspects of the course, and those with previous business experience appeared to more fully appreciate the interconnectedness of the business, engineering and computer science aspects of SOA than the class as a whole. The defined learning objectives were each met through use of a variety of course content that supported each of the outcomes. Each objective, and many of the outcomes, was supported by multiple instances of content with the goal of having always providing a practical and hands-on context to all theoretical concepts, and vice-versa.

5 Conclusions

This paper reports on the design and first offering of a course for graduate and upper level undergraduate students on the conjunction of computer science, engineering and business aspects of the emerging science of Service Oriented Architectures. Student learning in the course was significant and the objectives for that learning were all met. There was a variety of shortcomings and ideas for improvement expressed by the students, and by the instructor, and this feedback was used to design a second offering of the course which is being completed as this paper is written. We plan a follow-up study to compare the results of the second offering with the version reported in this paper, and to continue to incorporate feedback of all sorts into further development of the course.

Students particularly appreciated any attempts to provide real-world experiences, through topical lectures, large scale projects and guest speakers, and viewed such content as invaluable foundation for their current or future careers. Because the course was offered at the graduate level only due to course scheduling constraints, a number of options for adapting the course to upper level undergraduates were noted and will be incorporated when the course is offered at that level.

6 Acknowledgements

Support for the development of this course was provided by a Villanova Institute for Teaching and Learning (VITAL) Minigrant, with thanks to Dr. Carol Weiss for her suggestions and comments that helped with the initial offering. Faculty expertise in SOA modeling, design and implementation was gained through participation in a research project that was supported in part by the Air Force Materiel Command (AFMC), Electronic Systems Group (ESG) under contract number FA8726-05-C-0008. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of USAF, AFMC, ESC, or the U.S. Government. Thanks also are due to the faculty, staff and student researchers in the Villanova Center of Excellence in Enterprise Technology.

7 References

[1] Michael J. Carey. SOA What? Computer, 41:3, pp. 92-94, March 2008.

[2] Thomas Erl. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River, NJ, 2005.

[3] Mark D. Hansen. SOA: Using Java Web Services. Prentice-Hall, Upper Saddle River, NJ, 2007.

[4] Chang Liu. Enriching software engineering courses with service-learning projects and the open-source approach. In ICSE ’05: Proceedings of the 27th international conference on Software engineering, pp 613–614, 2005.

[5] Xuanzhe Liu, Yi Hui, Wei Sun and Haiqi Liang. Towards Service Composition Based on Mashup. 2007 IEEE Congress on Services, Salt Lake City, Utah, July 9-13, pp. 332-339, 2007.

[6] Eric A. Marks. Service-Oriented Architecture (SOA) Governance for the Services Driven Enterprise. Wiley Publishing, 2008.

[7] Michael Rosen, Boris Lublinsky, Kevin T. Smith and Marc J. Balcer. Applied SOA: Service-Oriented Architecture and Design Strategies. Wiley Publishing, Inc., June 2008.

[8] Rich Seeley. Enterprise mashups, SOA's killer app? SearchSOA, Jan. 29, 2008. <http://www.searchsoa.com>

[9] Mary Shaw. Software engineering education: A roadmap. In: A. Finkelstein (ed.): The Future of Software Engineering: Proceedings of the 22nd International Conference on Software Engineering, New York, NY, ACM Press, pp. 371-380, 2000.

[10] Melanie Trottman. US Air Logs Widespread Delays. The Wall Street Journal, Business section, March 5, 2007.

[11] Rayford B. Jr. Vaughn and Jeffrey Carver. The Importance of Experience with Industry in Software Engineering Education. Conference on Software Engineering Education and Training Workshops, 19th Conference on Software Engineering Education and Training Workshops, pp. 19, 2006.