A Reference Model for Master of Science Program in Services Computing

Liang-Jie Zhang[1], Zhixiong Chen[2], Min Luo[3], Jia Zhang[4], and Patrick C.K. Hung[5]
[1] *IBM T.J. Watson Research Center, Hawthorne, NY 10532, USA*
[2] *Mercy College, New York, USA*
[3] *IBM Software Group, USA*
[4] *Northern Illinois University, USA*
[5] *University of Ontario Institute of Technology, Canada*
{zhanglj, minl}@us.ibm.com, zxchen@ieee.org, jiazhang@cs.niu.edu, Patrick.Hung@uoit.ca

***Abstract*— Services Computing has become an increasingly important area in the IT and business sectors. In particular, Services now account for more than half of the economy in the United States and other countries. Numerous Services Computing-related degree programs and accreditation processes are being created. However, very few systematic guidelines exist for building graduate programs for Services Computing. In this paper, we present a reference model of the Masters Program in Services Computing for academic institutions and accreditation agencies as a relevant curriculum guideline. Specifically, the core and elective courses are introduced to help build the reference program. The inter-connections between core and elective courses are also illustrated to help create concentration programs based on the introducing sequences of the courses. Some practices of delivering Services Computing related courses and conducting accreditation application process are presented in this paper to help others more rapidly initiate the adoption process of the Services Computing curriculum.**

Keywords-Services Computing, Education, Curriculum, Reference Model, Master Program

I. INTRODUCTION

In response to the overwhelming and ever-increasing demands placed by the industry on trained specialists in modern service technologies, this paper aims to develop a reference model for the Master of Science (MS) program in Services Computing with the following three objectives: (1) to enhance the educational experiences and skills related to Services technologies, (2) to support the modernization and development of new curricula to teach Services Computing in a range of business and industry sectors, and (3) to facilitate collaboration between academia and industry to expand their training programs in Services Computing.

Services now account for more than half of the economy in the United States and other countries. To bridge the gap between business services and IT services, Services Computing has recently evolved as the foundational discipline to study how to leverage IT in order to successfully help people perform business services more efficiently and effectively [1]. It covers the science and technology of Services innovation and delivery that leverages information and computing technologies to be able to model, create, operate, and manage business solutions, scientific applications, and modernized services. Due to this reason, Services Computing has gained significant momentum from both academia and industry [2]. However, applying Services Computing technology to modernize the traditional services industry is now facing significant challenges. One major issue is the lack of skilled practitioners with the skills required to meet the increasing and urgent demands of the market.

Numerous Services Computing-related degree programs and accreditation processes have emerged in many countries. Unfortunately, very few systematic guidelines exist for establishing graduate programs for Services Computing. Therefore, the Technical Committee on Services Computing (TC-SVC) [10], within the IEEE Computer Society, is spearheading the compilation of experiences and efforts toward providing guidance to academic institutions and accreditation agencies about what should constitute a Services Computing curriculum.

Inspired from the Computing Curricula 2001 [8], IEEE TC-SVC started building Services Computing Curricula (SC) in 2007 [3]. Over the past four years, 14 key knowledge areas have been identified in the field and summarized as the Services Computing Taxonomy that has been used by the IEEE Transactions on Services Computing (TSC) [4]. Based on the key knowledge areas, TC-SVC has built Body of Knowledge (BoK) as a platform for the field of Services Computing to capture the latest advancements in the field.

Since 2007, TC-SVC has sponsored the Services Computing Curriculum Initiative (SCCI), the goal of which being to systematically integrate the SCCI practices and results into the creation of Services Computing degree programs suitable for accreditation processes. Following SCCI, in 2007, "Services University" [5] was launched at the IEEE Congress on Services (IEEE SERVICES) aiming to help Services Computing community members teach or learn the Services Computing discipline in a systematic way. It organized various programs to help educators teach Services Computing courses in collaborating universities, while helping industry practitioners obtain a comprehensive understanding of the subject.

Based on these extensive experiences, in this paper we propose a reference implementation of a Master of Science program for Services Computing and its curriculum in line with the SCCI requirements.

978-0-7695-4129-7/10 $26.00 © 2010 IEEE
DOI 10.1109/SERVICES.2010.107

As an initial proposal, this paper presents a MS degree curriculum that is comprised of 6 core courses and 10 elective courses based on the scope defined in [4]. A brief description of the core courses are presented in this paper as a reference. For the advanced training for university faculties in Computer Science, Software Engineering, Management of Information Systems (MIS), or closely related areas and senior IT professionals (Architects, Senior Software Engineers or Business Analysts), we can consolidate the 6 core courses into a one-week offering for special training, while focusing more on advanced and practical topics. We may also tailor some elective courses based on participants' individual needs and backgrounds.

The remainder of the paper is organized as follows: In Section 2, we depict the MS program overview with the outlines of the core courses, elective courses, and inter-connections. In Section 3, we present our preliminary experiences and guidelines of curriculum delivery and a degree program application process. Conclusions and future directions are described in Section 4.

II. PROGRAM OVERVIEW

A. *Courses Composition and Description*

Since the knowledge areas and sub-areas are reusable learning units [4] in the Services Computing Curriculum, they can be used to compose courses for different programs based on various requirements. There are several major usage scenarios for the key knowledge areas of the BoK. In industry, the BoK is used as a disciplined approach to drive the Services innovation agenda from research, development, and delivery perspectives. In academia, the BoK is used as the foundation for creating the program guidance for the Services Computing Curriculum. The key BoK areas are also used to form the taxonomy for the IEEE Transactions on Services Computing (TSC).

To illustrate this, for a course entitled "Introduction of Services System," we can select M.1, M.2, M.5.0, and M.5.5.2 [4] [9] to create this new personalized course for business professionals. We can add M.5.1 to this course for IT professionals.

There are a set of components in the course description for the Services Computing Curriculum. These include: course number, course title, instructor, abstract, prerequisites, syllabus, and additional notes. The course description should follow a university course catalogue and highlight the major topics and general expectations of the course. The prerequisites include the required courses, units, or background. The syllabus includes a list of units as defined in the SVC knowledge areas or sub-areas. The additional notes include comments about the course or pedagogical suggestions, assessment strategies as well as references, readings or textbooks.

Figure 1 presents the mechanism of the Course coding scheme. As shown in Figure 1, the course coding scheme includes the discipline name, the level of the courses, sequence number of the course in a degree program, and pedagogical approach (optional). In this case, SVC is used as the name of the discipline. It stands for Services Computing. The levels of each course could be categorized at level 0 for a core course, level 1 for an elective course, and level 2 for a special project. "01", shown in Figure 1, represents this course as the first course in the series. For the optional pedagogical approach perspective, "A" stands for a course that is technology centric. "B" represents course that is business centric. "C" means that a course can be used to teach both business centric students and technology centric students.

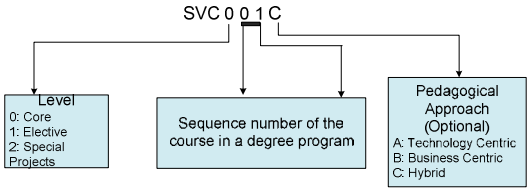

Figure 1. Course Coding Scheme

Figure 2 shows a course description template for each course in the Services Computing Curriculum. It includes key components, content for each component, and additional comments. The key components in each course description include Course Number, Course Title, Course Description, Prerequisites, Syllabus, and specific notes on this course.

Components	Content	Comments
Course Number		
Course Title		
Course Description (written in the style of a university course catalog, highlighting the major topics and general expectations of the course)		
Prerequisites (required courses, units, or background)		
Syllabus Covered (list of units covered as defined in the SVC body of knowledge)		BoK Areas-based Assessment Plan
Notes (Optional narrative section offering additional explanatory notes about the course. These may include goals, pedagogical suggestions, assessment strategies, references, and textbook)		

Figure 2. Components of A Course Description Template

The following section discusses the core and elective courses in the proposed reference model.

B. *Core Courses*

SVC001 - Fundamentals of Services Computing

From a technology perspective, Services Computing has evolved as the foundational discipline for the modern services industry to achieve higher reusability, scalability, interoperability, and cost efficiency. The Services Computing discipline addresses how to bridge the gap

between business and IT systems by leveraging IT and computing technology to help people design, develop, and deliver business services more efficiently and effectively. However, applying Services Computing technology to modernize the traditional services industry is now facing significant challenges. One major challenge is the lack of skilled practitioners able to meet the increasing demands from the growing services market. Addressing this challenge is one of the goals of this Master Program of Services Computing.

As the first starting course of this program, Fundamentals of Services Computing will introduce principles of services, services lifecycle, and services relationships. From an architectural thinking perspective, Services Computing adopts Service-Oriented Architecture (SOA) as a core architectural framework to provide a systematic approach for software system development and integration based on business logics. In SOA, functionalities are grouped around business processes and application packages as interoperable services. The technology known as "Web Services" is one of the widely adopted enabling technologies to realize SOA. Therefore, as examples, this course will also briefly introduce SOA and Web services technology.

SVC002 - SOA: Principles and Realizations

This course introduces students to a set of innovative research results and industry best practices around SOA. It presents students with various aspects of SOA including the latest concepts, services systems, modeling, publishing, discovery, invocation, composition, relationships, reference architecture, open source and open standard, governance, and enterprise architecture. After the completion of this course, students will have gained an overview and insightful understanding of the latest architectural principles and realization approaches.

SVC003 - Business Process Management and Integration

This course focuses on the fundamental concepts of business process management and integration across multiple services and enterprises. The integration can be categorized into two types: internal and external integration. Internal integration includes all the integration aspects of services within one enterprise. External integration covers all possible service integration patterns across multiple enterprises. The enterprise infrastructure has to provide the capability to discover and integrate services as well as enabling federated security mechanisms, services monitoring, and management.

This course covers various practices, standards, and technology in the design and realization of business process modeling, management, and integration. This course includes a series of hands-on technical exercises in business process application development. Through business process application development, students are able to apply the knowledge and skills learned from the lecture materials into a practical business process transformation environment.

SVC004 - Service Management: Security, Privacy, and Assurance

Service Management covers one of the most critical factors in making Services Computing successful in both promoting it, and, more importantly, bringing true business value to the enterprise. It covers the security, privacy, and assurance issues of Services Computing. This course also covers challenges, prevailing technology, industry standards, and best practices.

This course includes six second-level knowledge areas: General Topics, Access Control in Services Systems, Security Enablement in Services Systems, Privacy Management, Quality of Service and Service Assurance in Services Systems. The general topics include Security Concerns of Service-Oriented Solutions, Privacy Concerns of Service-Oriented Solutions, and Trust in Service-Oriented Solutions, Quality of Services measurement and management, and finally, processes and supporting technology for service assurance.

SVC005 - Services Computing Consulting Methodology

This course concentrates on the fundamental concepts of Services Computing consulting methodology that can be used by a practitioner to conduct enterprise transformation based on IT systems. This course introduces basic frameworks of using analysis methods to address an enterprise's pain points, identify gaps, and define IT service initiatives to support the real business needs by leveraging the latest Services Computing technology. This course covers three key knowledge areas: Enterprise Modeling, Enterprise Performance/Portfolio Management, and entry-level Business and IT Consulting Methodology.

The course also covers the key concepts used in the consulting methodology: Balanced Scorecard, Component Business Model, Enterprise Architecture, Enterprise Project Management, Service-Oriented Business, and IT Consulting Methodology.

SVC006: Services Delivery

This course concentrates on the basic concepts of services delivery and engineering practices. The core body of knowledge areas included in this course are: Services Delivery Platform and Methodology, Business Grid and Cloud Computing. The key course modules include: Outline, Concepts, Platform, Reusable Services, Methodology, Outsourcing-based delivery, Global delivery, Collaborative delivery, Software as a Service (SaaS), Services As Software, Operational aspects of Cloud Computing, Architectural aspects of Cloud Computing, Information assurance in Cloud Computing, and Project Management in services delivery.

In addition to the course description, each core course may have several hands-on-labs. In this paper, we will use SVC006 as an example course to illustrate two hands-on-labs.

SVC006-Lab1: Services Mashup and Delivery over the Internet

The expected deliverables of this lab are 1) identified available services from the public domain for your project; 2) Services mashup solution for aggregating those identified services based on certain business logic; and 3) Functional and non-functional capability report of your lab project. The students need to complete this lab project within 6 weeks.

SVC006-Lab2: Provisioning in Cloud Computing

This hands-on-lab requires students to use open source software or vendor-specific products to create a virtualized Cloud computing environment for provisioning resources. The students can choose one of the following scenarios to complete the lab project: Infrastructure as a service, Software as a service (Middleware sharing), or Business application as a service.

C. Elective Courses

We have also built a list of the following ten elective courses to focus on specific knowledge areas in a deep-dive manner. All the elective courses are coded with SVC1xx, where "xx" represents the sequence number.

SVC101. Web services
SVC102. Information Assurance in Services Computing
SVC103. Cloud Computing
SVC104. Enterprise Architecture
SVC105. Information Architecture and Transformation
SVC106. Services Value Assessment
SVC107. Patterns and Model Driven Development
SVC108. Open Source for Services Engineering
SVC109. Services Quality
SVC110. Capstone Research Project

The detailed description for all the elective courses will not be included in this paper due to space limitation. As an example, this paper includes SVC103 to show the description style is the same as the core course.

SVC103 - Cloud Computing

This course concentrates on the basic concepts, architecture, business models, and case studies of Cloud computing. The core body of knowledge area included in this course is Business Grid and Cloud Computing (M.8), Services Delivery Platform and Methodology (M.11). The key course modules include Outline, Concepts, Cloud Computing Platform, Cloud Computing Infrastructure Management, Platform As A Service, Software As a Service (SaaS), Business Process As A Service, and Application As A Service, Security and Privacy in Cloud Computing, Design-for the Cloud, Design-in the Cloud, Pay-per-usage pricing model, and Cloud Computing Deployment. Two hands-on labs (SVC103-Lab1: Cloud Computing Infrastructure (hardware setup and middleware enablement; SVC103-Lab2: Business Cloud), and an exam are also included in this course offering.

For other special-purpose courses, we will use SVC2XX as the coding mechanism for the rest of the courses posted on the Services Curriculum wiki [6].

D. Course Inter-connections

In this section, we will briefly discuss the inter-dependencies among the core and the elective courses for the Master degree program. Figure 3 depicts the major dependencies and helps to facilitate the sequencing process. Five categories of focused areas can be identified and used to guide students in their selection of courses or focuses.

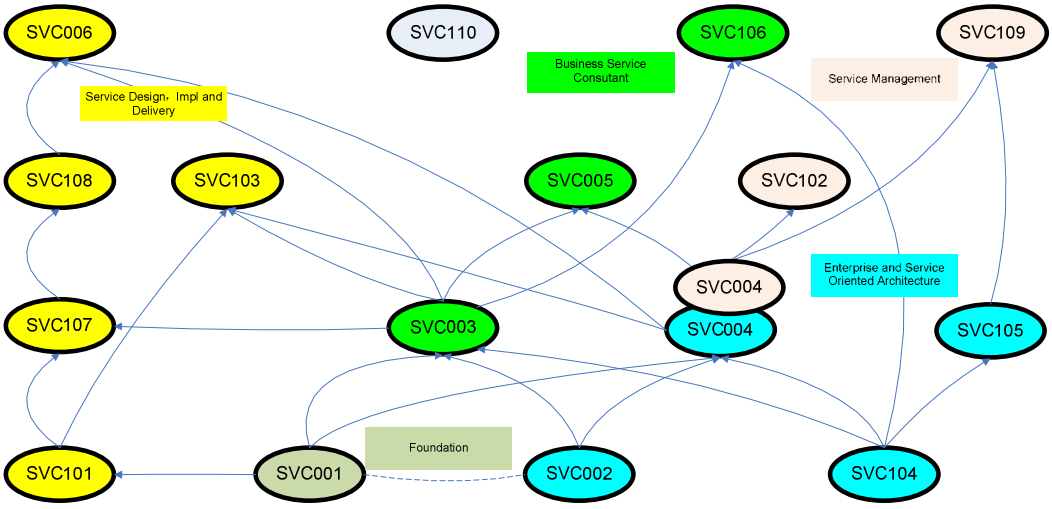

Figure 3. Interconnections between Courses

Foundation

As one can understand, the foundational course for Services Computing is SVC001, with possible additional contents from SVC002.

Enterprise and Service Oriented Architecture

One of the fundamental problems with not-so-attractive IT Return of Investment (ROI) is that most enterprise systems and applications have been built in an ad hoc way for the past several decades. This explains why those systems are very difficult and expensive to enhance or integrate, making them unable to adapt to the ever-changing business requirements.

The MS in Service Computing will make a conscientious effort to introduce "architecture" as the cornerstone for the proposed program that aims to help students and practitioners fully understand the importance of establishing a sound architectural foundation for enterprise systems and applications, how to address concerns of various key stakeholders of their specific perspectives or interests using various architecture views, and how Services Computing would become an effective mechanism to facilitate an architecture-centric process for the whole lifecycle of system development. SVC002, SVC004, and the electives SVC102 and SVC106 will accomplish this objective.

Business Service Consulting

This sequence consists of SVC003, SVC006 and SVC106. One of the assumptions or prerequisites here is that students should have taken a business requirement analysis and modeling class, presumably in their junior or senior year in college, or they will be required to study on their own in order to take SVC003. They must also be able to understand business process management and integration, how business requirements can be captured and effectively reflected or represented in business processes. SVC005 will then discuss the consulting methodology for Services Computing, utilizing the lessons learned from large real projects and proven best practices to effectively engage with customers at all levels.

The ultimate purpose for business (Services) consulting is to demonstrate how business processes could be improved with Services Computing. How to prove this to the CxOs has been a challenging problem for both internal IT organization and consultants. SVC106 will be dedicated to business value assessment with practical and proven set of processes and metrics.

Service Design, Implementation and Delivery

This sequence will focus on how to analyze, design, implement, and deploy well-specified, open standards-based services using mostly open-source technologies and commercial tools on the market. Model-driven business development methodology will be discussed and demonstrated through case studies and projects, so that students will gain hands-on experience following those proven methodologies, and be able to solve real world problems through the use of these technologies. This sequence mainly consists of SVC101, SVC107, SVC108, and SVC 006.

Service Management

A critical inhibitor for Service Computing presents more challenges on information security, privacy, integrity, and quality of services with the new service-oriented paradigm. This sequence will first provide a broad overview of the issues, current technologies, and their applicability and limitations, evolving open standards, and best practices in SVC004. SVC106 and SVC109 will then focus on more in-depth studies of service/information assurance and service quality.

Special Projects

Based on students' interests, special projects can be selected and will focus on solving some of the specific issues related to Service Computing.

Selected sequences can also be served as specialized training or certification offerings for IT professionals who want to become SMEs in some of those special areas.

III. PROGRAM DELIVERY PRACTICES

A. *Strategies for Curriculum Instructors*

A successful SVC curriculum instructor should have sufficient knowledge and experience of the SVC techniques, while also having strong grasp of the overall picture of the field.

It will be helpful for instructors to study the SVC taxonomy and thoroughly understand the relationships between various directions and concepts. A combination of theory and hands-on experience will help course organization. Because the SVC discipline evolves quickly, it is important for instructors to keep up-to-date with the knowledge and development in the field, and regularly review and update course materials as necessary. Since the SVC field covers a broad range of areas, specializations toward specific types of Services Computing should be based on instructors' previous or current research in related areas.

There are several key points that the educators should keep in mind when delivering Services Computing courses. First, Services Computing courses should be taught in a way to seamlessly incorporate computing and engineering perspectives. Second, students should be trained to become service consultants with a business view in mind. They have to bear in mind that SVC projects and solutions aim to deliver user-centered services. Third, the traditional computer science teaching philosophy, learn-by-doing, is a good strategy to teach Services Computing courses. Fourth, it is important that students learn how to select proper and

up-to-date tools in order for them to gain experience using the tools to solve practical problems.

B. Strategies for Curriculum Designers

SVC curriculum design should be driven by outcomes. Keeping in mind the expected outcomes will help to focus on the designed courses and materials to ensure they are being taught in a proper manner and with an appropriate level of depth. In addition, not only is it important to keep a certain amount of content with technical and mathematical depth in the curriculum, it is also equally important to allow instructors to add some advanced materials.

Content should be organized in a way to ensure that simple materials will be covered earlier in a course to prepare students for studying more comprehensive techniques. Meanwhile, core concepts will be revisited and reinforced throughout the courses. From our experience, we found it highly useful to engage in practical materials early in a course (a problem-driven technique), so students can be increasingly motivated and be more willing to study related techniques to solve the issues. Some traditional courses in Computer Science, Software Engineering, Management of Information Systems, and Business could be migrated to the corresponding courses proposed in this MS program.

Cases studies showing how the Services Computing techniques can be applied to advance other application domains and disciplines (e.g., telecommunications and scientific computing) will be helpful. This will give students experience applying the Services computing technologies and methodologies to solve practical problems after they join the workforce.

C. An Example Degree Application Process

Since a degree application process is tedious, slow and overwhelming, it is suggested this program be started within existing departments first. Once the program begins to grow rapidly, creating a new department or school may be a natural choice. Unless well planned, it is quite possible that the process can drag on for as long as two years. Based on our experience in successfully applying for a similar Master and Bachelor degree program [7] in New York State (USA), we would like to share our application process with the community.

The application process was delivered in several stages. These include: a) brainstorming, b) market analysis, c) academic merit analysis, d) facility and faculty capacity analysis, e) curriculum design, f) at least two levels of curriculum committee reviews, g) provost and president recommendations, h) trust review, i) state education department evaluation, j) major revision and public response period, k) program launch and l) program review.

a) Brainstorming

During the brainstorming stage, we need to discuss the need for such a degree program. We will ask ourselves if it is more beneficial to have a concentration under some existing degree program/s. We will investigate the impact to other existing degree programs. The rationale behind this approach is that we do not want to sacrifice other degree programs in order to start a new one. Lastly, we need to understand what kinds of expertise the current faculty members have.

b) Market analysis

An analysis of supply and demand in relation to this competitive market is necessary for a degree program such as this since the success or failure of a program depends on the needs of the industry.

We can conduct a series of surveys about the demand for specific knowledge and skills, such as by distributing questionnaires to local or nationwide industrial and government agencies. We can also utilize existing survey results from big consulting companies or magazines. Job market web site surveys can also be used as indicators of the demands.

In general, we need to investigate the surrounding colleges and universities for similar programs. Having a similar program in the surrounding areas can convince us of the demand. However, these institutions can also be opposed to having a similar degree program in their backyards, when the state education department asks for response in the final application evaluation process. We need to prepare a rebuttal to such claims.

On the other hand, if there is no similar program in the surrounding area, it does not mean that we should wait. If we perform an analysis, we can determine factors such as the strength, weakness, opportunity, and threats (SWOT) in the market so we can make the right decision.

c) Academic Merit Analysis

This academic program should not be considered a fashion show. It should contain a long-term goal and objective that will deliver value long into the future. Therefore, we should determine the core knowledge foundation of the program, the expanded knowledge scope and depth, as well as required practical skills.

d) Facility and faculty capacity analysis

Once we determine the knowledge base, we need to develop the program in a systematic way. If needed, convincing faculty members to work with the domain experts in the field, or attend the Services University program [2] is a good starting point.

During the capacity analysis process, we need to decide what facilities such as labs, reference text, books, and journal requirements are needed. We also need to decide if a new line is needed. We also need to determine an approximate dollar amount to be used in developing and maintaining such a program, which will be used by the trustees for final approval.

e) Curriculum design

The main part of the application process is the curriculum design. It requires a substantial amount of study and research. We need to develop admission requirements; pre-requisite lists, core courses, elective courses and final thesis or program options.

Designing a course is a long-running process. It requires very careful evaluation including knowledge items to be covered, skills to be developed, text selection, bibliography, outcome objectives and assessment plan. In general, we need to prepare three piece documents for each course development: course application form, course syllabus and course outline as a sample. Different institutions may ask for more materials.

Designing a capstone project or thesis is also a challenge. Most master programs have an option to do a master project for three credits or thesis for six credits. It will operate very smoothly if we list requirements in as much detail as possible.

The next task in this stage is to draw several possible course sequences and determine the appropriate amount of time required to complete them. These different course sequences can serve the diversities in the student body.

Again, curriculum design is a very long and tedious process. If it is not created and maintained in a sufficient manner, it will impact the program later on and eventually slow down the growth of the program.

f) Curriculum review

In general, the curriculum needs at least two levels of review: department or school level curriculum committee review, and college or university curriculum committee review.

During the departmental review, many insightful comments pertaining to course design and contents will be made. We need to revise the curriculum carefully to receive an approval. In addition, we need to understand the time frame for such a review process since there are at most, three meetings in each semester in many schools. If we miss this opportunity, we may have to wait for another semester.

During the college curriculum committee review, scholars with varying backgrounds tend to make many generic comments due to their different perspectives. Again, we will need to revise the curriculum and be aware of the timeframe for such a review, especially when it requires a second reading.

g) Provost and President Recommendation

The next stage is the recommendation by the Provost and then by the President. Their decisions are made based on the college mission, objectives and budget. This should not be a problem if we communicate with them from the very beginning of the program application.

h) Trustee Level Review

Trustee-level review focuses on long-term financial commitment and the impact on the college. The curriculum itself needs little revision. Again, the timeframe is very important.

i) State Education Department Application draft and submission

Once we have been authorized by the college to proceed, we need to move onto presenting the actual program application to the state education department. In this stage, we need to get the actual application form and fill in all required information. The form needs to be signed by the Provost or the President before submission.

j) Revision based on state evaluation and public response

The college should maintain a close contact with state evaluators, discuss the application with them, and respond to their concerns actively instead of waiting for final responses. Once we receive the comments and decision from the state, we need to make changes accordingly if required.

Usually in this stage, the department will forward a letter to the surrounding colleges about the new program, and ask for their opinions. We need to be prepared to defend the program in the event there are any objections to the program's implementation.

k) Program launch

Program launch should be carefully planned. We need to educate admission personnel about the new program, prepare a market plan for student recruitments, build new lab facilities and library resources, and staff for new courses being offered.

A new program is not like a well-established program. It requires a lot of explanation and marketing. Without implementing a proper market plan, the program's survivability will be low. Marketing materials include a well-designed Web site, marketing brochure, and contact information. We notice that many students research information directly from the Internet, so this will be used as a prime source for information.

l) Program Review

Program launch is not a done deal, it needs to be reviewed regularly and changes must be made as required. The review should be based on objective evaluation such as course evaluation and instructor evaluation. It should reflect actual student needs as well as changes in the field.

IV. CONCLUSIONS

In this paper, we presented motivations for creating a reference model for the Master of Science in Services Computing program. The course coding scheme and 6 core courses were identified and introduced. We defined 10 elective courses for various focuses, which can be applied to build various concentration programs based on the course interconnections presented in this paper. We also presented a practice of conducting an accreditation process, which could be leveraged to build Services Computing degree programs.

For future directions, we believe adding some criteria to measure quantitative learning outputs would enhance topics for further exploration.

ACKNOWLEDGMENT

We would like to send our thanks to the participants of various panels, birds-of–feather sessions, and plenary talks on Services Curriculum initiative at ICWS, SCC, CLOUD, SERVICES, APSCC, ECOWS, and COMPSAC for sharing with us their inputs and comments on various versions of this reference model.

Dr. Hong Cai contributed to some discussions on this master program. We also want to thank the reviewers for their valuable review comments.

REFERENCES

[1] L.-J. Zhang, J. Zhang, and H. Cai, Services Computing, Springer and Tsinghua University Press, 2007.

[2] L.-J. Zhang, Services Computing: A New Discipline, International Journal of Web Services Research (JWSR), http://www.servicescomputing.org/jwsr, 2004.

[3] L.-J. Zhang, C. K. Chang, Towards Services Computing Curriculum, Congress on Services - Part I, (SERVICES). IEEE, pp.23-32, 6-11 July 2008, Hawaii, USA

[4] L.-J. Zhang, Introduction to the Body of Knowledge Areas of Services Computing, IEEE Transactions on Services Computing, Vol.1, No.2, April-June, 2008, pp.1-15.

[5] L.-J. Zhang, A Services University, IT Professional Magazine, March/April 2009, 11(2), pp. 60-62

[6] Services Curriculum wiki, http://www.servicescurriculum.org, Sponsored by TC-SVC, IEEE Computer Society.

[7] Mercy College Master and Bachelor degrees in Information Assurance and Security, https://www.mercy.edu/academics/school-of-liberal-arts/department-of-mathematics-and-cis/ms-in-information-assurance-and-security/

[8] Computing Curricula 2001, Computer Science, Final Report, The Joint Task force on Computing Curricula, IEEE Computer Society and Association for Computing Machinery, IEEE Computer Society Press, December 15, 2001.

[9] IEEE Transactions on Services Computing (TSC), http://www.computer.org/tsc.

[10] The Technical Committee on Services Computing (TC-SVC), IEEE Computer Society, http://tab.computer.org/tcsc.