Systems Oriented Architecture, Unified Process Life Cycle, and IS Model Curriculum Compatibility: Meeting Industry Needs

Frederick Kohun
kohun@rmu.edu

David Wood
wood@rmu.edu

Joseph Packy Laverty
laverty@rmu.edu
Computer and Information Systems, Robert Morris University
Pittsburgh, PA 15219, USA

Abstract

Model Information Systems (IS) Curricula have circulated since the 1980's . While computing education has continued to grow and evolve, recently there has been the introduction of yet another model curriculum-- the 2006 Information Technology (IT) Model Curriculum. Furthermore with the 2001 beginning of ABET-CAC accreditation for IS and the 2006 ABET-CAC accreditation for Information Technology the role of model curricula with respect to standards has increased in importance. This paper addresses the compatibility of such model curricula to incorporate specific industry based design and implementation methodologies. The methodology discussed in this context will be the IBM's implementation of Services Oriented Architecture (SOA) within the Unified Process Life Cycle. This methodology was chosen for two reasons: 1) there is relatively high demand for graduates with familiarity and competence with this methodology, and 2) SOA is the maturation of object-oriented systems development strategies which require a more adaptive system life cycle.

Keywords: Systems Oriented Architecture, Unified Process Life Cycle, IS Model Curriculum, ABET-CAC, Accreditation

INTRODUCTION

The emergence of Object-Oriented Programming Languages, XML, and web services has attempted to address limitations of the historical silos representing applications, platforms and data by promoting the creation of reusable software modules into a Service Oriented Architecture (SOA). As SOA concepts are applied in a business context, the relationship with a Systems Development Model is an important consideration. IBM implementation of Services Oriented Architecture is implemented through the Unified Process Life Cycle Model, which is based on Agile Software Development Methodologies.

SOA was chosen for two reasons: 1) There is relatively high demand for graduates with familiarity and competence with this methodology, and 2) SOA is exclusively directed toward the object oriented environment currently practiced in today's web based world.

In its software development suite, IBM has integrated SOA within the framework of the Unified Process Life Cycle that has incorporated business analysis principles. The impact on business expenditure is highlighted by Darryl Taft, "Impact 2007 is IBM first customer event focusing on SOA, which analysts say is a $160 billion market opportunity. And IBM touts that the company has successfully delivered on more than 4,500 SOA engagements for customers." (Taft, 2007) "Hewlett-Packard's $4.5 billion deal to acquire Mercury Interactive is the right fit for service-oriented architecture, analysts said." (Taft, 2006)

The Spiral model of systems development, the foundation of the unified process, is mentioned in the IS2002 Model Curriculum. Object Oriented approaches are also included. The 2006 IT model curriculum also includes Object Oriented methodologies, although systems development is not a major emphasis. Currently, IT organizations in industry recognize SOA as an implementation extension of Object Oriented development concepts and the Unified Process Life Cycle as the implementation extension of the Spiral model. However, neither SOA nor the Unified Process Life Cycle are explicitly included in either the ABET-CAC standards nor the available model curricula.

BACKGROUND AND RELEVANT LITERATURE OF AN SOA ENVIRONMENT

The following sections briefly identify and define the major concepts that frame the basis of the SOA methodology.

Unified Process

Iterative and Incremental adaptations were introduced to overcome the limitations of the waterfall approach to SDLC. Probably the most famous adaptation, the spiral model, was introduced by Barry Bhoem in his 1986 article A Spiral Model of Software Development and Enhancement. (Boehm, 1986) Unlike a Top-Down Approach which divides complex projects into smaller, more easily managed modules, the spiral model uses iteration to simplify the design and development process. (Satzinger et al, 2005, p40,41)

In 2002, a new iterative system life cycle model, named the Unified Process Life Cycle (UPLC), was proposed by Rational Software Corporation, a division of IBM. The four phases of Unified Process Life, i.e., inception, elaboration, construction and transition were repeated for each iteration phase. UPLC went beyond simple changes in phase titles. There is no equivalent Elaboration Phase in the waterfall SDLC. The Elaboration phase integrates the learning benefits from each iteration by refining requirements, scope, cost and estimates. For example, the project requirements, scope and estimates are not fixed, they evolve. (Satzinger, 2005, p46)

Bergandy (2006) provides an excellent framework for the Unified Process Life Cycle that includes the four phases: inception, elaboration, construction, and transition. Projects never really terminate or begin, but the system is assumed to constantly reinvent itself. In his framework, he lists both the objectives and the life-cycle milestones for each phase. Most versions of SOA are based on the Unified Process Life Cycle

Service Oriented Computing

“Service Oriented Computing is the paradigm that [utilizes] services as fundamental elements for developing applications” such that “services provide a distributed computing infrastructure for both intra- and cross- enterprise application integration and collaboration.” (Papazoglou and Georgakopoulos (2003) “Service Oriented Computing” as quoted by Connolly (2005)) This definition seems to be one of the earlier foundations in the application of the Services Oriented Computing concept. It is refined and taken one step further by Lim and Jong (2006). They say: “One can define a service-oriented paradigm as one that utilizes services as fundamental building blocks for developing applications. It can be thought of as being reincarnated from the concept of ASP (Application Service Provider), another popular software concept of the 90’s where an application is served over a network.” (Lim and Jong, 2006)

One of the most recent treatments of Service Oriented Computing in the IS spectrum was in the context of web services based on the IBM Service Oriented Architecture (SOA) by Connolly (2005). Connolly’s work succinctly defines the basis of this new design paradigm by identifying the solution to dealing with the “twin problems of integration complexity and reuse”. Furthermore, he defines the primary difficulties associated with application integration as the large number of interfaces required.

In addition, shortly thereafter, Lim and Jong (2006) go further to demonstrate how SOA technologies can be introduced into an IS curriculum. They discuss the nature of web services and SOA and their inclusion in a first level course in an IS curriculum.

Yet another work by Rob (2006) compares the traditional "Waterfall" Systems Development Life Cycle (SDLC) to the Object oriented unified Process approach to systems development which provides the foundation for the SOA approach.

Varying operational definitions of SOA

The required features of an SOA are not agreed upon, and have evolved over time to encompass software, hardware, data, and business processes. The definitional characteristics of Service Oriented Architecture that are listed in Table 1 by along with a sample of the most cited sources are as follows:

1) Modularity of Service Components

Information Service is encapsulated in a separate module. Modules incorporate the data being presented, and any functional routines needed.

2) Reusability and Flexibility

Separate information service modules need to be usable by many different applications. They exist in a library accessible to all software relevant, and can be used and reused by multiple programs concurrently.

3) Model-Based Development

Software development should follow a prescribed set of processes and standards, ensuring consistency across firms and applications.

4) Platform and Interface Independence

Information services need to be separated from the physical implementation of the information. Thus, the same service should work with MS Windows, Linux, Macintosh, or any other operating system. Many of them are delivered across the web, potentially to different devices, screens, phones, etc.

5) Interfaces Publishable and Discoverable

Each module must reveal its public properties and description to any querying application, which means it must adhere to standard WWWC interface requirements

6) Data Interfaces

Each module must be able to recognize standard data structures for other web and service data, especially to read XML and SOAP standard syntax.

7) Business Integration Emphasis

SOA should not only include software development and its processes, but also Business process reengineering, or redesigning the business to take advantage of Information Services characteristics.

8) Web Applications

SOA incorporates web services and their delivery across the internet.

9) Distributed Capabilities and External Partners

Information Services need to cross corporate boundaries, and be usable by other firms, supporting a multi-institutional supply chain.

10) Event Driven Triggers

SOA needs to respond to events from the environment, both within a firm and across the internet, and not be controlled only hierarchically from applications within one firm or program.

While these characteristics taken from the literature form the basis of a holistic definition of Service Oriented Architecture, Table 1 shows the gaps for a unified definitional consensus :

ABET CAC ACCREDITATION STANDARDS, IS CURRICULUM, AND SOA,

The intent of the curriculum category is to combine "professional requirements with general education requirements and electives to prepare students for a professional career in the information systems field, for further study in information systems, and for functioning in modern society. The professional requirements include coverage of basic and advanced topics in information systems as well as an emphasis on an IS environment. Curricula are consistent with widely recognized models and standards." (Kohun and Wood , 2003). For most programs this means that the IS2002 or the CIT2006 model curricula form the basis of the programs.

These curricula are broken into goals and courses which are descendents of the waterfall SDLC approach to software development and not explicitly the SOA Model.. To see if they are flexible enough to embrace SOA, we looked at the core courses required at one mid-size university with an ABET-CAC accredited information systems program and attempted to map the (core) curricula to the SOA and Unified Process Model. The curricular mapping is listed in Table 2.

RESULTS

DISCUSSION

Pieces of the SOA and the unified process are incorporated wherever objects are taught, but they did not form the foundations of the core of the curriculum. Object Oriented Systems Analysis texts and modern Database texts which treat objects introduce some of the unified process concepts. Traditional Systems Analysis refer to rapid design techniques, but do not concentrate on the granularity of objects required for information services.

Current model curricula are now three generations behind industry practices using SOA. The curriculum relies on Waterfall SDLC while industry has, in many cases, moved to SOA, UPLC, and IBM (business) SOA. Object Oriented Systems Analysis texts have adopted UPLC (Spiral Model), but it still is rare

The ABET notion of requiring an Information Systems Environment creates a "silo," separating business courses from IS courses. To incorporate SOA with business process reengineering would mean a much more integrated delivery of business and organization MIS concepts.

With current industry practice and market trends, perhaps we should rethink the composition and flexibility of the foundational model curriculum

REFERENCES

Boem, B. (1986). "A Spiral Model of Software Development and Enhancement." ACM SIGSOFT Software Engineering Notes, August, 1986.

Connolly, R. (2005). "A Funny Thing Happened on the Way to the Form: Using Game Development and Web Services in an Emerging Technology Course." Information Systems Education Journal, 3(38). Chicago: AITP.

Crawford, C.H., Bate, G.P., Cherbakov, L., Holley, K., & Tsocanos, C. (2005). "Toward an on-demand service oriented architecture." IBM Systems Journal, 44(1). Retrieved 6/10/2007 from Website: http://www.research.ibm.com/hjournal/sj/441/crawford.htm.

Elmasri, R. and Navathe, S. (2007). Fundamentals of Database Systems, 5th edition. Boston: Addison Wesley.

IBM (2007). Retrieved 6/10/2007 from Web site: http://www-306.ibm.com/software/solutions/soa/.

Kalata, K. (2005). Introduction to ASP.NET 2nd ed. Boston: Thomson Course Technology.

Kendall, K. &Kendall, J. (2008). Systems Analysis and Design, 7th ed. Upper Saddle River, NJ: Prentice Hall.

Kohun, F., Wood, D. (2003). "The ABET-CAC Accreditation Experience – Intent and Reality – The Information Systems Perspective." Information Systems Education Journal, 1(1).

Lim, B.B.L., Jong, C.J. (2006). "Integrating Service-Oriented Paradigm into Introductiory IS Courses." Proceedings of ISECON 2006. Dallas: ISECON.

OASIS (2007) Retrieved 6/10/2007 from http://en.wikipedia.org/wiki/OASIS_SOA_Reference_Model

OMG (2006). Retrieved 6/10/2007 from Website: http://colab.cim3.net/cgi-bin/wiki.pl?OMGSoaGlossary#nid34QI.

OMG (2007). Retrieved 6/10/2007 from website: http://www.omg.org/.

Opengroup (2006). Retrieved 6/10/2007 from http://opengroup.org/projects/soa/doc.tpl?gdid=10632.

Rob, M.A. (2006). "Dilemma Between The Structured and Object-Orineted Approaches to Systems Analysis and Design." Journal of Computer Information Systems, 46(3). Stillwater, OK: IACIS.

Satzinger, J.W., Jackson, S.D., & Burd, S.D. (2005). Object-Oriented Analysis and

Design with the Unified Process. Boston: Thomson Course Technology.

Schwalbe, K. (2007). Information Technology Project Management, Fifth Edition. Boston: Thomson Course Technology.

Shelly, G. Cashman, T., Woods, D., & Dorin, W. (2007). HTML Comprehensive Concepts and Techniques, 4th ed. Boston: Thomson Course Technology.

Taft, D. (2006) "HP's Mercury Buy Boosts SOA Play." retrieved September 27, 2007 from http://www.eweek.com /article2/0,1759,1994189,000.asp

Taft, D. (2007) "IBM Shows Impact of SOA." retrieved September 27, 2007 from http://www.eweek.com/ arcticle2/0,1895,2134619,00.asp

Techweb (2007). Retrieved 6/10/2007 from website: http://www.techweb.com/encyclopedia/definetерm.jhtml?term=soa.

Webopedia (2006). Retrieved 6/10/2007 from web site: http://isp.webopedia.com/TERM/S/ Service_Oriented_Architecture.html.

Zak, D. (2004). Microsoft Visual Basic .Net: Reloaded. Boston: Thomson Course Technology.

Appendix

Table 1. Essential Concepts in SOA

SOA Characteristic Emphasized	(Webopedia 2006)	(Techweb, 2007)	(OMG, 2007)	(OASIS . 2007)	(Lim & Jong, 2006)	(Connolly, 2005)	(Crawford, 2005)
Modularity of Service Components	No	No	No	Yes	Yes	Yes	Yes
Reusability and Flexibility	No	No	No	Yes	Yes	Yes	Yes
Model-based Development	No	No	Yes	Yes	Yes	No	Yes
Platform and Interface Independence	Yes	No		Yes	Yes	Yes	Yes
Service Interfaces (W3C -- Publishable and discoverable interfaces)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data Interfaces	Beyond XML and SOAP	XML, SOAP	No	No	Yes	No	Yes
Business Integration Emphasis	No	No	Yes	Yes	No	No	Yes
Web Applications	No	Yes	No	No	Yes	Yes	Yes
Distributed Capabilities and External Partners	No	No	No	Yes	Yes	Yes	Yes
Event Driven triggers	SOA 2.0 Combines SOA and Event Driven Architecture	No	No	No	No	No	Yes

Table 2. SOA and the Unified Process in typical textbooks.

Course	Textbook	SOA Inclusion	Unified Process Inclusion
VB.Net	Zak	None	None
Operating Systems	Tanenbaum and Woodhull	System Services only	None
Web Development	Shelly, Cashman, Woods, Dorin	None	None
Systems Analysis	Kendall and Kendall	Includes UML and the IBM Rational Suite	Object oriented Methodology, Agile Modeling and prototyping,
Advanced Systems Analysis	Satzinger, Jackson, and Burd, Object-Oriented Analysis and Design with the Unified Process. 2005	Includes UML and objects.	Agile Development
Network / Data Communications	N+ text	None	None
Advanced Web Page Development	Kalata	XML Web Services SOAP	None
DBMS	Elmasri and Navathe	Object Databases and interoperability (ODMG standard). Doesn’t refer to SOA	Includes Rational Rose and UML
Project Management	Schwalbe (2007) Information Technology Project Management	None	None