Communications of the Association for Information Systems

CAIS

Impact of Service-Centric Computing on Business and Education

J. Leon Zhao
University of Arizona

lzhao@eller.arizona.edu

Michael Goul
Arizona State University

Sandeep Purao
Penn State University

Padmal Vitharana
Syracuse University

Harry Jiannan Wang
University of Delaware

Abstract:

Service-centric computing is one of the new IT paradigms that are transforming the way corporations organize their information resources. However, research and teaching activities in the IS community are lagging behind the recent advances in the corporate world. This paper investigates the impact of service-centric computing on business and education. We first examine the transformative impacts of service-centric computing on business and education in the foreseeable future. Then, we discuss opportunities and challenges in new research directions and instructional innovations with respect to service-centric computing. We believe that this article will serve as a good starting point for our IS colleagues to explore this exciting and emerging area of research and teaching.

Keywords: application development, business services, education, service management and computing, service-centric computing, service-oriented architectures, systems analysis and design, standardization, Web services

Volume 22, Article 16, pp. 295-310, March 2008

I. INTRODUCTION[1]

The past few years have seen a dramatic development of service-centric computing[2] and related technologies such as Web services, service-oriented architectures, and business process automation. According to Gartner, by 2011, 63 percent of products in the software infrastructure market and 56 percent in the software application market will support Web services and Web 2.0 technologies. Consequently, more business applications will be delivered as IT services. Service-orientation represents the most recent evolution in the corporate computing paradigms, which can be traced to several generations starting with monolithic, client-server, object-oriented, component-based, and finally, service-oriented.

The service-oriented approach has emerged by building upon component-based techniques that leverage object-orientation and client-server architectures, as well as business process automation. However, the complexity of service-centric computing brings forth tremendous challenges and exciting opportunities in corporate information management. Up until now, there is little research examining the impact of service-centric computing on business and education. As such, there is a great opportunity for the IS community to take a leadership role in this research area. In essence, we are in a pre-paradigm state with respect to longitudinal impacts, so initial research perspectives need to be synthesized and critiqued — particularly given the transformation promises of service-centric computing. This paper summarizes and integrates the insights of the authors and discusses how educational institutions should interpret and help to shape this paradigm shift in enterprise architectures and infrastructures as the corporate world adopts and adapts service-centric computing.

While the jury is still out on the ultimate success of service-centric computing, we believe that service-centric computing will impact business in several ways. First, service-centric computing makes it possible to better align IT with business strategies and improve the quality of IT services in an unprecedented manner. Second, the business of application development, maintenance and evolution will change dramatically as model-driven application development takes root and more and more web services are published and subscribed to. Third, service-centric computing will enable more effective business process automation that can integrate internal and external business operations more flexibly. In this paper, we outline the basic concepts about the business impacts of service-centric computing.

The IS curriculum must change in response to these shifts. First, IS students must understand service-based application development and process management, and how it is premised on web services standards. Second, IS students must learn new tools for service-oriented architectures and business process modeling so they can meet the demands of corporations in their deployment of service-centric computing. Third, IS students must appreciate changes to system development practices, which must now incorporate componentized applications and leverage service integration across the organizational value chain.

The IS community, therefore, needs to reposition itself to meet the challenges of service-centric computing and prepare the next-generation of professionals, architects, and developers to traverse the wide spectrum of service-centric IT solutions. This requires new computational and managerial frameworks that will not only impact corporate computing practices but also lead to pedagogical initiatives within and outside the IS discipline.

The remainder of the paper is organized as follows. Sections II through V develop several viewpoints on various fundamental aspects of service-centric computing. Section VI integrates these viewpoints to generate a set of propositions that may drive research and education on service-centric computing. Finally, Section VII concludes the paper by calling on the IS community to make more concerted efforts to contribute to knowledge creation and pedagogy development as related to service-centric computing.

[1] This paper is based on the AMCIS 2007 Panel on Impact of Service-Centric Computing on Business and Education. Nonetheless, a significant collaborative effort after the panel has extended the paper well beyond what was discussed during the panel.

[2] We use the term service-centric computing to cover numerous related terms such as service-oriented computing, Web-service computing, and service science, management, and engineering.

II. SERVICE-CENTRIC COMPUTING AS A DISRUPTIVE TECHNOLOGY

Service-centric computing is a new term coined to refer to a set of computational techniques including Web services, service-oriented architectures, and service science, management and engineering [Zhao et al. 2007]. Service-centric computing is based on the concept of service orientation under which various computing services work together to provide the needed computational capability in an enterprise. Although a service resembles a system component in the paradigm of component-based software development, it is different in that a service in service-centric computing must be able to function independent of other services, thus not any system component can be called a service. The goal of service-centric computing is to enable the enterprise to grow and redesign gracefully.

Service-Oriented Architecture at the Core of Service Centricity

At the core of service centricity is service-oriented architecture (SOA). SOA was first proposed in 1996 by Gartner analysts Roy W. Schulte and Yefim V. Natis. They specified SOA as " a style of multi-tier computing that helps organizations share logic and data among multiple applications and usage modes. " Since then, SOA has been described in many ways. Huhns and Singh [2005] describe SOA from a technical perspective as the system architecture that enables the creation, search, assembly, and utilization of Web services to support various business processes in order to achieve various business goals. According to DMReview.com, a service-oriented architecture is (a) a collection of services that communicate with one another; (b) the services that are self-contained and do not depend on the context or state of other services; and (c) the services that work within a distributed system architecture.

Gartner [2005] describes SOA from the business perspective. Their definition begins by describing services as independent, self-contained, reusable business functions (such as credit checking) or infrastructure functions (such as user identification). Services can be combined and orchestrated to automate complex business processes. The main objective of SOA, then, is to enable a more agile, flexible and standardized approach to designing, developing and deploying functionality that is often scattered throughout established IT systems. Many organizations, thus, regard SOA as an architecture that enables business agility through the use of common services.

Honey [2006] at Kaiser Permanente advocated that successful implementation of SOA requires the organization to emphasize reuse of IT resources. First, the organization must set policies and goals on how to achieve levels of reuse of its information resources. Second, it needs to move from strategies that are narrowly focused on programs to ones focused on producing and integrating reusable services across the entire enterprise. Third, the organization should alter its software development and capital planning processes to look for reuse opportunities as a core IT mission. Fourth, it must change the corporate culture through a combination of executive recognition and incentive programs that strongly reward reuse. Finally, the organization needs to develop services that may be used by multiple organizations, not just by local users, while putting appropriate service level agreements in place.

The SOA Consortium

Organized and managed by the Object Management Group, the SOA Consortium is a SOA advocacy group committed to helping the Global 1000 successfully adopt SOA by 2010 (www.soa-consortium.org). The SOA Consortium believes that SOA adoption is a key enabler for the 21st century enterprise and achieving the benefits of SOA requires significant changes for both IT and business executives. It is interesting to note that the consortium indicates that SOA is about IT integration and productivity, rather than business agility, which presents an opinion different from Gartner on the breadth and depth of the impacts of SOA. Nevertheless, the SOA Consortium agrees with Gartner that SOA is based on process-driven IT services and requires integrated business and IT management.

A recent article by the SOA Consortium reported the changing role of IT from "cost-cutting and creating operational efficiencies" to "contribution to revenue growth" [SOA Consortium 2007]. As such, the role of the CIO has expanded to participation in corporate strategy development. Specifically, CIOs are identifying ways technology can spur business innovation and value generation. These viewpoints are supported by recent research from Gartner Group and the Economist Intelligence Unit: "According to recent CIO polls from research firm Gartner Inc., 50 percent of CIOs surveyed said they now have duties outside of core technology, such as helping to craft corporate strategy."

On the Disruptive Effects of Service-Centric Computing

As a computing paradigm, service-centric computing is different from many other paradigms such as structured programming, object-oriented programming, and component-based software development. The difference lies in both the goal of computing paradigm and the scope of the impact.

As shown in Figure 1, SOA as the core concept of service-centricity is to enable tight integration of business strategy and IT strategy. None of the prior corporate computing paradigms claimed such a far-reaching goal. Although the switch to object-orientation in software development challenged older programmers and project managers and required retraining and retooling, it did not affect the boardroom significantly. In contrast, the shift to service-centricity will require the full participation of business managers and executives in order to achieve dynamic integration of business and IT functions at the strategic level.

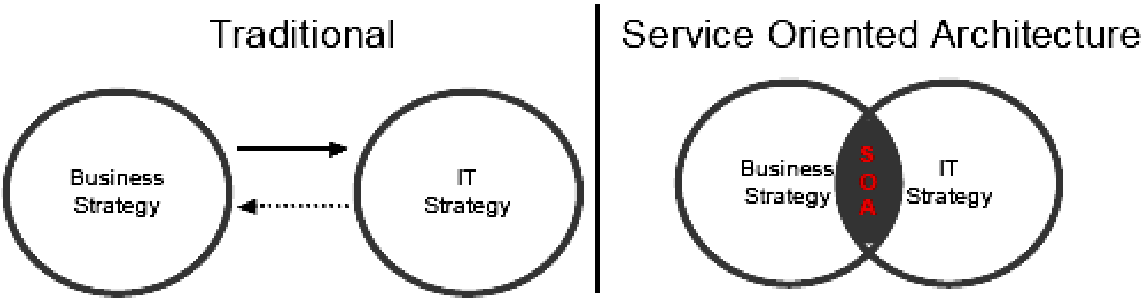

Figure 1. SOA Enables Tight Integration of Business and IT [Adopted from BEA 2005]

The disruptive effect of service centricity will alter how corporations function computationally and what business managers do in strategic planning, assuming that the alignment of business and IT strategies must occur as a two-way street. That is, business and IT alignment needs the cooperation of business executives and IT managers. As pointed out by Paul Horn [2005], service centricity requires companies " to merge technology with an understanding of business processes and organization and to understand how that capability can be delivered in an efficient and profitable way. "

As a result, the shift in the corporate computing paradigm toward service-centric computing will require universities to revamp their curricula in both business and IT education. Paul Horn [2005] put this issue bluntly by noting that "there's a shortage of skills where they're needed the most — at the intersection of business and IT. As companies build more efficient IT systems, streamline operations, and embrace the Internet through wholesales changes in business processes, a huge opportunity exists."

The rest of this article will shed light on how the IS community can develop a research agenda as well as a strategy for designing/revising courses to adopt service computing and management in order to contribute to the knowledge gap and to prepare our students for corporate needs in service-centric computing.

III. THE TRANSFORMATIONAL IMPACTS OF SERVICE-CENTRIC COMPUTING

There are two converging drivers serving to reinforce the necessity for aligning business perspectives with organizations' IT investments. First, the globalization of the economy has brought significant focus on three primary business strategies: labor arbitrage, local adaptation, and aggregation [Ghemawat 2007]. All three strategies, particularly aggregation, rely on an extensive set of technical architecture capabilities. Aggregation requires advanced standardization such as that related to the efforts of the SOA Technical Committees of standards organizations like OASIS.

As a second driver, markets are changing competition from company vs. company to value chain vs. value chain. Thus, for a company to compete in an increasingly global context, agile and adaptive business environments are an imperative. Further, these business environments demand transparency coupled with changeable business processes and strategies. Service-centric computing is intended to enable this transformation.

The Role of Service-Centric Computing in Transforming Global Competition

Service-centric computing will need to support IT in its evolution to a third generation of process automation practices, software and solutions. This generation will "reclaim degrees of freedom lost to proprietary solutions that were designed to operate within four walls" [OASIS ebSOA TC 2006]. Service-centric computing will enable this transformation by relying on a process-centric, semantics-based perspective as discussed in the following.

The basic building block of SOA is the service. A service is a business task or activity that is realized through technology. It is important to note that tasks and activities are also important components in a business process; hence, a service in SOA can be thought of in terms of the business tasks or activities the software sources, supports, or implements. This means that SOA is the framework that composes services to enable e-business in accordance with business objectives. From this angle, SOA is defined through its transformation of business requirements into the technology needed to deploy and execute business processes in SOA enabled infrastructure.

Further, that framework should provide models, methodologies, best practices, standards, reference architectures, and run-time architectures that are needed to provide guidance and support through all SOA design, development, production and deployment phases. Such a framework is substantially different than historical application lifecycles, agile development methodologies and application-oriented computing infrastructures.

The SOA framework must also differ from historical approaches due to the need for supporting more transparency, particularly with respect to value chain vs. value chain competition. Transparency is required in two areas. First, communications within a value chain must be consistent and standardized by the semantics used to convey intensions among collaborating partners. Communicating from SOA enabled organizations to other SOA enabled organizations will require new “federation” standards. Second, there must be metrics associated with the most important value chain activities as communicated through those semantics. For example, with design activities taking place in the value chain, there will be semantics associated with designs such as “release drawing,” etc. Similarly, there may be cycle time metrics, quality metrics, co-authorship metrics, etc. as related to “release drawing”. Thus, service-centric computing intended to support value chain vs. value chain competition will require the type of transparency needed to enable collaborating organizations to continuously improve and innovate.

Service-Centric Computing and Educational Transformation

The next generation of IS graduates must be comfortable with business processes, the mapping of process tasks to compositions of software-based services, and the emerging design, development and deployment paradigms for reference architectures described above. While service-centric computing is aiding in the transformation of business and value chains, services-based educational delivery is concomitantly emerging. It is highly likely that “Reusable Learning Objects” (RLOs) will be the preferred paradigm for delivering e-learning services. RLOs for service-centric computing are already being designed by industry and used in some business schools [e.g., IBM Academic Initiative 2007]. This approach allows for agile course development and delivery, and it foretells a major transformation to the role of an educator from curriculum developer and deliverer to curriculum discoverer and integrator [Lang and Zhao 2000].

Technology Alignment

Three areas of emerging technology are consistent with the business-oriented, service-centric computing approach described above. First, SOA is a key enabler of value chain linkage across federated infrastructures, assuming the emergence of standards for facilitating this linkage [e.g., OASIS ebSOA TC 2006]. Second, semantics are the business action language that will facilitate collaboration across the federation, and those semantics will give rise to value chain metrics to support value chain vs. value chain competition. Master Data Management (MDM) is the key enabler for translating value chain semantics for internal consumption and vice versa. Finally, Active Data Warehousing (ADW) is the key enabler of providing the relevant and timely data to business processes that operate across a federated landscape. Companies like Intel and American Express are already working towards this technological alignment [Goul 2007]. The current challenge is to operationalize this trifecta of technologies, i.e., SOA, MDM, and ADW, to align with relevant standards as they emerge and to replicate successful strategies across value chains.

A Grand Challenge

Service-centric computing challenges will be related to evolving standards and best practices such as those discussed earlier. It is clear that the measure of success of service-centric computing will not be in service “ reusability” (as has been the case with many “component-centric ” initiatives); rather, the impact must be transformational and concomitant with the business drivers shaping modern global competition. *One grand challenge for service-centric computing will therefore be to enable transformation in inter-organizational collaborations that have remained under-addressed.* For example, there are many solutions involving standards, semantics and metrics for supply chain collaborations that can serve as the initial test bed for research-based methods and technologies, but a true test will be to extend the transformational capabilities to collaborating service organizations. How should the trifecta of technologies (MDM, SOA and ADW) be arranged for this context? What are the needed semantics? How about the most important metrics? These and other questions suggest a broad array of needed research in the service-centric computing area. In short, they imply a grand challenge to those researchers and practitioners interested in or working in collaborating service organizations.

IV. BUILDING SERVICE-BASED INTEGRATED SYSTEMS

One domain, where service-centric computing is seeing immediate application is the design and deployment of integrated systems that are built from existing and new services. A recent SEI report [Brownsword et al. 2006] states: “We have crossed a threshold where most of our large software systems can no longer be constructed as monoliths … They are now constructed as groups of interoperating systems (as systems of systems).” This nomenclature reemphasizes integration concerns that continue to be the main driver behind several advancements related to interoperability. These integrated systems tend to be large-scale, intentionally scalable, support constantly evolving work practices, and can require significant collaboration among vendors and organizations to build [Watts 2006]. Next, we identify and discuss four key challenges related to building service-based integrated systems.

Standardization of the Web Services Substrate

The first challenge refers to standardization of the Web services substrate that underlies service-based system integration solutions. Unlike some other domains, standardization is not an academic exercise nor one intended to merely cause Web services to improve. For Web services, standardization is vital for the very existence and realization of interoperation. The Web services standards stack has been evolving in an anticipatory manner [Cargill 1989] to address the complex and interdependent concerns that the standardization space offers. The core standards related to publishing, finding and binding have been developed as separate, yet interdependent standards with more recent activity directed to the development of concerns such as reliability and security, among others [WSDL 2001; UDDI 2005; SOAP 2003]. These so-called *WS-** concerns [Erl 2005] along with the more established Web services standards represent a standards stack that has been investigated by a few authors. One recent attempt represents use of the Language Action perspective to develop a reference framework, which provides pointers that potential designers and managers may use to assess existing standardization efforts [Umapathy and Purao 2007].

A related concern is the process of designing these standards. Although a number of prescriptive approaches have been suggested for standards development [see, e.g. McCallum 1994; Moen 1998]; and circulars have been issued [OMB 1998] and laws passed [HR-1086 2003] to recognize the public policy environment and due process requirements [ANSI 2005]; the actual processes of standardization continue to defy standardization. Standardization processes are now carried out under the auspices of voluntary standards development organizations such as W3C, OASIS and others in a manner that involves a significant design component [Mitra et al. 2005]. Concerns for organizations affected by standardization include questions such as whether and how to participate in standards development activity. A useful meta-theoretical perspective in this regard is offered by Fomin et al. [2003], who suggest design [Simon 1981], sense making [Weick 1995] and negotiation [Latour 2005] as a recursive process, where closure in each is a necessary condition for each level of recursion. Archaeological studies of standardization processes for Web services standards are also beginning to shed light on intricacies of these processes and how organizational participants engage in the standardization activity [Varili 2003; Mitra et al. 2005]. Both “standards” and “standardization” of the Web services substrate, thus, represent ongoing research activities and opportunities that researchers are beginning to explore with significant implications for business and education.

Designing and Evolving Service-Based Integrated Solutions

The second challenge relates to our ability to build these service-based integrated systems. Prior work suggests that although we can conceive these large-scale integrated systems we have trouble building them [Brownsword et al. 2006; Watts 2006]. The record of accomplishment of projects undertaken to build system integration solutions continues to be sub-par [Bajaj et al. 2005; Charette 2005]. While it is tempting to dismiss past failures as a “human problem” [Wysocki 1998], it is imperative that we develop knowledge-based approaches to address the problem of architecting, designing and evolving service-based integrated systems. Unfortunately, techniques to develop system integration solutions are still mired in traditional approaches that are dominated with a “design first, deploy next” mindset [Archer 1984; Boehm et al. 2005]. There is, however, a growing recognition that building integrated systems is an ongoing process [Schmidt 2004] that is likely to involve: harvesting and/or identifying services, designing how the services will collaborate with one another in the context provided by cross-functional processes, and continuously evolving these solutions to adapt to changes in the environment.

Although lifecycle approaches are being proposed that encompass and make clear the transition from services to processes and from processes to service-based integrated systems, much work still remains to be done to develop actionable and comprehensive approaches that may allow us to overcome the traditional disconnect between design and maintenance. One potential approach to overcoming this disconnect being debated in recent academic research is the use of interactions among Web services as a building block for designing and evolving service-based system integration solutions [Barros et al. 2005; Umapathy et al. 2003; Peltz 2003]. Several elements of a comprehensive approach appear to be available as proposals in different disciplines such as message exchange patterns, collaborative agents for workflow decomposition, and monitoring mechanisms. Combining these in effective ways to form comprehensive approaches to build and deploy service-based integration, however, remains a significant concern for research and practice.

Managing Risks in Multi-Vendor Projects

The third challenge relates to project and risk management practices related to large-scale, multi-vendor projects that are necessary to implement and deploy service-based integration solutions. Because of their scale and distributed nature, such integrated solutions simply cannot be designed as monoliths; instead, they are evolved over time by connecting legacy applications and new software with a foundation of services that allows interconnections. Projects to implement these solutions, therefore, tend to be large-scale, long-duration and highly complex making them highly susceptible to failure. Understanding, tracking and mitigating risks in these projects are difficult because of the distributed locus and emergent nature of these projects, which further intensifies the already difficult problems of low task observability [Kirsch 1996; Kirsch et al. 2002], distorted communications [Snow and Keil 2002], reluctance to report bad news [Tan et al. 2003] and escalation propensities [Keil and Robey 1999] that plague systems design projects. A number of research streams can contribute to developing risk mitigation approaches including but not limited to data fusion, software metrics and information market mechanisms. Useful approaches to risk mitigation can, therefore, include providing early warnings of impending project failures, and capturing pooled assessments of risks from the distributed participants. Few research initiatives address these concerns.

Learning To Build Service-Based Integrated Solutions

The fourth challenge relates to pedagogy. A number of characteristics of building service-based integration solutions require a rethinking of strategies used for educating computing professionals. While the traditional vision of the IT professional as a toolsmith has served computing curricula and education efforts reasonably well for the last few decades [Brooks 1996], the role of the IT professional itself is now changing — from that of a toolsmith to that of a participant in a multi-disciplinary team of diverse professionals [Edens 2000]. A successful IT professional must be self-directed, participate effectively in a team, be aware of information technology standards necessary for loose-coupling and interoperability [Watts 2006; CRA 2006], and be able to work at the boundary between information technology and work practice. This represents a daunting list requiring multiple epistemologies for learning [Nilsen and Purao 2005].

V. REQUIREMENTS ANALYSIS AND SYSTEM DESIGN IN SERVICE-BASED DEVELOPMENT

Service-based development involves two key phases: service fabrication and application assembly. In service fabrication, services are built while in application assembly, applications are built using these preexisting services. Issues relevant for research in service fabrication include the abstraction of the domain in terms of services, service granularity, service classification and coding for the purpose of storing them in a repository, among others. On the other hand, issues in application assembly include matching requirements with services, locating corresponding services, and assembling them. Most gains can be achieved when services are built for an entire domain and then using those services to build applications for that domain.

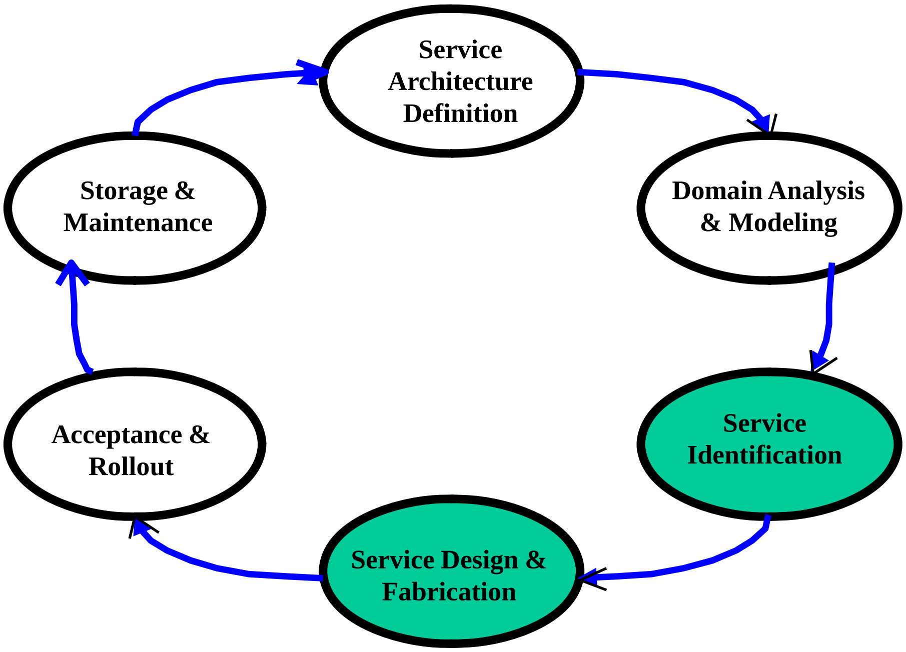
Service Architecture Definition
Domain Analysis & Modeling
Service Identification
Service Design & Fabrication
Acceptance & Rollout
Storage & Maintenance

Figure 2. Service Fabrication

Figures 2 and 3 graphically illustrate current practice in service fabrication and application assembly. As shown in Figure 2, the corresponding phases in requirement analysis (RA) and design during service fabrication are: (a) service identification and (b) service design and fabrication. Here the services are identified as a result of domain analysis which are then designed and fabricated. On the other hand, as shown in Figure 3, the corresponding phases in RA and design during application assembly are: (a) requirements definition where requirements for the proposed application are defined, (b) service search and selection where the services matching requirements are searched and selected, and (c) application is assembled using these services.

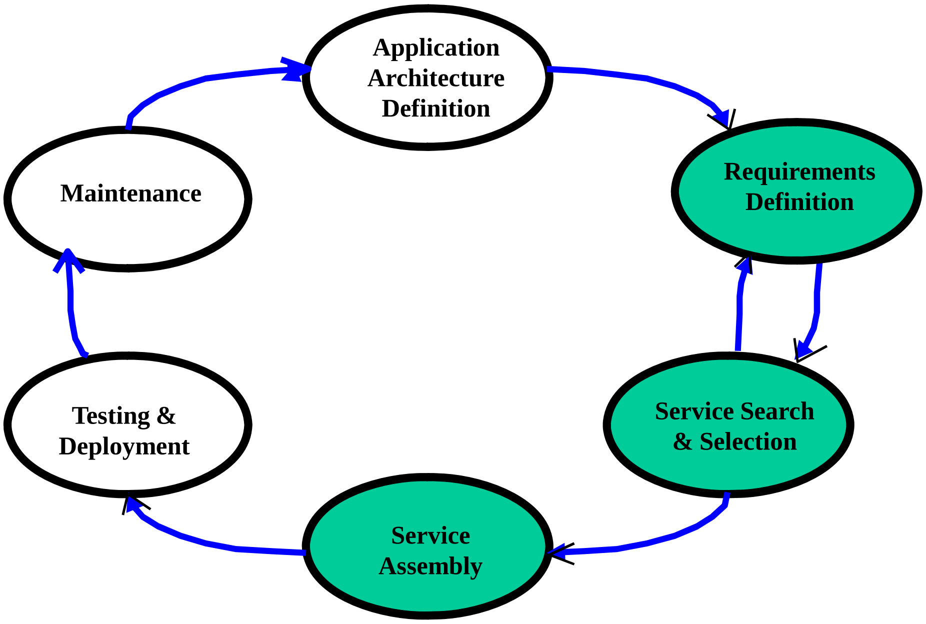

Figure 3. Application Assembly

In traditional development, requirements analysis and design play a key role in application development. The emerging paradigm based on existing services will fundamentally alter these two upstream stages of application development. Service based applications are built by composing services possibly sourced from various providers archived and described in some repository. When these repositories contain domain encompassing services, requirements analysts could learn about the domain by examining related services. Because analyst's knowledge of the business domain is critical to RA success [Pitt and Brown 2004], future research could examine if and how such a repository could help the analyst acquire the domain knowledge.

Furthermore, in traditional application development, derived requirements are used as a blue print later in the design stage. In the service paradigm, design takes a different view. Design is likely to turn into an exercise in matching user requirements with pre-existing services. Future research should focus on how the service paradigm alters requirements analysis and design stages of application development. When a given requirement does not perfectly match the functionality of existing services, a new service might have to be built anew, which can then be placed in the repository for future use. Moreover, in cases where some requirements do not match existing services, user group could be given the opportunity to alter their requirements based on the cost-benefit analysis of using existing services and building of services anew. As illustrated in Figure 3, in the emerging paradigm, RA and service search and selection tasks go hand in hand.

Lately, there has been a considerable discussion on the need for agility as organizations compete in a global landscape. It has been argued that the service paradigm could serve as the foundation for enterprise agility [Zhao et al. 2007] by enabling organizations to more readily adjust to demands from customers and competitors. By adopting service based development, organizations are expected to be more agile vis-à-vis traditional methods of application development [Jain and Vitharana 2008). Enhanced agility could be realized at both RA and design phases. From using service repositories to helping analysts garner knowledge of the application domain during RA to building applications from pre-fabricated services, the new paradigm provides the ground work for organizations to become more agile. Nonetheless, there is little empirical research that has examined how service-based development affects enterprise agility.

VI. OPPORTUNITIES AND CHALLENGES

The previous sections of this article present various fundamental viewpoints, which should be of value to the IS community as they ponder about the issues related to service-centric computing. In this section, we synthesize these viewpoints to formulate a number of propositions that highlight the opportunities and challenges in service-centric computing and their impact on business and education.

Proposition 1: Service centricity ensues naturally when an area becomes more commoditized.

According to the 2007 research survey of *InformationWeek* 500 executives, 70 percent of the companies have widespread Web services applications deployed in their IT infrastructures and 37 percent of them have SOA implemented and accessed by more than half of their workers [InformationWeek 2007]. Service-centric computing, including Web services and service-oriented architectures, no longer represents the hype and buzzwords people used several years ago; they have been fully commoditized and have helped thousands of companies to gain competitive edge and tangible benefits. We argue that a services focus is what naturally ensues when an area, more specifically the information systems area, becomes more commoditized. Standards and standardization of the Web services substrate, therefore, represents a substantial opportunity for research as well as for businesses and education.

In general, the software industry has been an example of how increasing standardization and commoditization have reduced risks thereby facilitating cost effective outsourcing. Growth in service-centric computing will mean that outsourcing can be realized at a much more granular level. Software itself has no intrinsic value, but the context within which the software delivers a business service does have value. Since high-quality business services often command premium margins, it is reasonable to expect that services-centric computing will bring more and more value to business as providers compete through differentiation. This will create an exciting new business model that radically differs from software licensing, shrink-wrap, etc.

Commoditization will also bring opportunities to innovate. For example, aggregating commodity components, such as constructing new services from aggregations of services, represents how commoditization can actually enable new opportunities. Those opportunities might not have been possible prior to commoditization, and such innovations can provide further differentiation and are likely to result in even better business services. It is no surprise, then, that standards to facilitate Web service aggregation (composition) have already become embraced and entrenched in service-centric computing. In summary, while commoditization might conjure dire images for some involved in the IT discipline, service-centric computing actually represents a natural evolution that provides an opportunity for providers to differentiate in providing more and more business value. This bodes well for the IS discipline as a whole.

Proposition 2: Service-centric computing is more than making the IT department a service organization and represents a fundamental transformation that requires a change of mindset.

Service-centric computing should not be confused with currently prescribed management approaches for operating an IT department as a service organization, e.g., Information Technology Infrastructure Library (ITIL). As noted earlier, service-centric computing refers to a set of computational techniques including web services, service-oriented architectures, and service science, management and engineering [Zhao et al. 2007].

Service-centric computing represents the latest evolution of the architectural styles and related technologies for enterprise systems. It reflects the need for higher levels of abstraction beyond specific platforms and has resulted in a fundamental transformation that requires a change of mindset [Zimmermann et al. 2005]. In particular, we need to change our mindset from various perspectives including business, development, research, and education.

From the business perspective, the globalization and highly dynamic business environment require companies to adapt and respond to changes efficiently in order to maintain their competitive advantage. Service-centric computing provides a solution to achieve the required enterprise agility at the system level. Managers need to change their mindset to think about how to integrate services marketing into process redesign and service-level performance management, how to achieve more flexible and reliable supply chain by collaborating with their partners via open-standard-based system integration, and how to ensure the proper alignment between the service-oriented IT infrastructures and overall business strategy.

System architects need to change their mindset when designing service-oriented enterprise architectures by choosing open standards over proprietary specifications, maximally leveraging existing assets by wrapping them as standard services versus building applications anew, and investigating how to design and deliver IT services in the face of inter-organizational alliances and service-oriented partnership.

From the development perspective, building applications in the new service paradigm requires a significant change in the mindset of analysts and developers [Brown et al. 2005]. Analysts need to focus more on how to match business requirements with existing services rather than design systems from scratch, and developers need to learn how to build service-based integrated systems by composing and orchestrating existing services.

From the research and education perspective, there are many issues confronting the IS community, which include: (1) Establish a cooperative research agenda that bridges the gap between traditional computing and new business processes associated with co-creation and co-production of IT services; (2) Sort out the terms and vocabularies in service-centric computing that have led to confusions; (3) Develop and revise IS curriculum to supply the required knowledge and skills for the service paradigm.

Proposition 3: Service-centric computing leads to a greater emphasis on integration of business and IT.

As companies expand their business into new markets through mergers, acquisitions and consolidations, achieving the ability to respond to organizational changes with integrated, flexible, and responsive systems falls on the shoulders of IT. Service-centric computing offers the key technologies, such as Web services and service-oriented architectures, used in the mainstream IT solutions to transform existing enterprise infrastructures to achieve higher level of agility [Erl 2005]. The success of the business relies more and more on the successful implementation of service-based IT solutions, which makes the integration of business and IT much more important than ever before.

Service-based IT infrastructures can mitigate the risk of changing business, technology, and legislative environments and contribute directly to business success in several ways. First, leveraging existing system assets is one of the key benefits of service-based IT solution. By reusing legacy systems, service-centric computing allows organizations to best use their past investments although it requires emphasis on service-orientation to ensure flexibility, scalability and adaptability. Thus, IT becomes the direct contributor to profits rather than its previous role as a profit facilitator. Second, once the service-oriented infrastructures are in place, the development cycle time of new applications could be dramatically reduced, resulting in new products getting to market faster. For instance, according to a recent InformationWeek survey, after implementing SOA based infrastructure, Wachovia could build non-trivial solution on top of their existing services in four to twelve weeks versus the six to nine months it used to take [InformationWeek 2007]. Therefore, taking time to build well-architected services leads to huge saving in future solution developments, which aligns with business strategy very well. Third, risk of system development needs to be significantly mitigated by appropriately leveraging service-based, multi-vendor distributed projects, which present the opportunities as well as threats to the design and implementation of successful solutions.

Proposition 4: Service-based development is the process of building applications from a set of services.

In the service paradigm, an application is derived from an amalgam of related services that satisfy corresponding business requirements. When building applications, the service paradigm and the associated vocabulary provide the analysts and developers an appropriate medium for tackling the problem. The new paradigm enables organizations to build both inter- and intra-organizational applications. More importantly, it facilitates the integration of inter-firm and intra-firm applications that could be built on separate platforms [Jain and Vitharana, forthcoming, Singh and Huhns 2005].

Requirements analysis and design are more intertwined than conventional development [Vitharana et al. 2007]. When reusable services are in place, analysis and design become an exercise in service selection and application assembly. Hence, once requirements are identified, they are matched against available services. For requirements that are not fulfilled by existing services, new services need to be built. The user group could be informed of the availability of existing services and their corresponding costs. As appropriate, the user group could be given the option to alter their requirements based on cost factors.

Service-based development is the process of building applications from a set of services. These services could be built anew based on a specific need or could be re-factored from legacy systems. Nonetheless, instead of building services piecemeal, generating a set of services for a particular domain such as human resource function or supply chain management could prove most useful [Vitharana et al. 2007]. In doing so, the domain knowledge and corresponding functionalities need to be abstracted in terms of loosely coupled services that then could be used to compose applications for the domain. Existing methods in domain analysis provide a starting point for abstracting domain artifacts in terms of services. Vitharana et al. [2004] proposed an approach for using the domain object model to derive components by considering high level managerial goals (e.g., reusability, ease of assembly) and technical features (e.g., coupling, number of components). Once services are built, they need to be coded and placed in a repository for subsequent search and retrieval. Building and evolving new services and service-based integration solutions can, then, proceed in tandem.

Proposition 5: Service-centric computing has led to new opportunities for new IS courses and new service-related modules in existing courses.

Institutions offering programs in information systems need to offer new or revamp existing curricula to take the changes brought by service-centric computing into account. Specifically, the aforementioned changes in the requirements analysis and design stages of the service-based development warrant a new mindset in approaching these two early stages of systems development. Therefore, we need to change the traditional system analysis and design course to include contents on business domain analysis, service identification and design, service discovery and matching, and service composition and orchestration. In the service-centric computing paradigm, more and more applications are built by leveraging existing system components via standard service interfaces instead of being developed from scratch as taught in most existing programming courses. Thus, we could incorporate into existing programming courses contents on how to transform existing system components into standard services, how to publish services to service registry, and how to build service-based integrated systems by composing services.

These changes shift the focus of the courses away from low-level software development and move it more towards business-oriented application development and integration. Consequently, these changes may be of particular interest to IS programs in business schools where students take business courses as part of their curriculum but lack new technical knowledge and sophistication to cope with the changes due to service-centric computing. The demand on business analysts, business developers, and business integrators has spurred recently, and newly designed IS courses with service-related modules will equip the students with necessary knowledge and skills to compete in the service-oriented business environment. This demands not only changes in course contents but also pedagogical practices.

Service-centric computing also creates many opportunities for new courses. For example, one aspect is to use information as services, where master data scattered within heterogeneous application silos across the enterprise can be wrapped as services, accessed via standard service interfaces, and integrated to provide more accurate and holistic view for decision making. This knowledge is in high demand while companies are leveraging service-centric computing to transform their information management, which has not been covered in existing database courses. Industry has also noticed the knowledge shortage in the realm of service-centric computing and has been working with academics to build new programs and curriculum. IBM has helped many universities to establish programs related to service science, management, and engineering (SSME), such as UC Berkeley, Carnegie Mellon University, NC State, to name a few. Programs such as IBM Academic Initiative, Oracle Technology Network, and Teradata University Network provide the latest service-oriented systems, tools, and courseware to educational institutions free of charge. The IS community must exploit these resources to guide educational initiatives for service-centric computing to the next level.

VII. CONCLUSIONS

This article examined an emerging enterprise computing paradigm called service-centric computing and explained why service-centric computing is a disruptive technology and how it might transform enterprise computing. We also discussed how to build service-based integrated systems and how requirements analysis and system design might change under service-based development. Finally, we presented fundamental viewpoints and a synthesis in the form of a number of propositions that outline opportunities and challenges in this area of research and teaching. As might be expected of an area that is large and diffused with a set of perspectives, an overarching theoretical perspective is difficult to pin down to appreciate and understand the impacts of service-centric computing on research, education and business. In lieu of such an overarching perspective, we have opted for a set of propositions that can provide interesting starting points for our colleagues in the IS discipline and tentative identification of foreseeable trends for businesses engaged in or interested in deploying service-centric computing. Clearly, there are additional propositions that may be derived from the perspectives on service-centric computing. The propositions presented should, therefore, be seen as a necessarily incomplete set that reflects our collective belief in the key directions that need more immediate investigation.

We believe that this article marks an important signpost for research and teaching on service-centric computing in the IS community. That is, this article underscores that service centricity has become an imperative topic of discussions for the foreseeable future. To achieve the impacts and directions for research and teaching outlined in this article will take collective efforts by many IS colleagues. We can safely predict that the IS community will make significant contributions with respect to the adoption of service centricity in boardrooms as well as classrooms. We hope the viewpoints and propositions provided in the article will be instrumental in judging how we can channel our energies in making these contributions worthwhile.

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers, who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The authors of the Web pages, not CAIS, are responsible for the accuracy of their content.

4. The author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

ANSI. (2005). “ANSI Essential Requirements: Due Process Requirements for American National Standards,” http://www.itl.nist.gov/biometrics/Requirements0405.doc.

Archer, L. B. (1984). “ Systematic Method for Designers, ” in *Developments in Design Methodology*, N. Cross (ed.) John Wiley, pp. 57-82.

Argyris, C. (1976). *Increasing Leadership Effectiveness*, New York, NY: John Wiley.

Bajaj, A., D. Batra, A. R. Hevner, J. Parsons, and K. Siau. (2005). “ Systems Analysis and Design: Should We Be Researching What We Teach? ” *Communications of the Association for Information Systems*, vol. 15, pp. 1.

Barros, A., M. Dumas, and A. Hofstede. (2005). “Service Interaction Patterns,” In Proceedings of the 3rd International Conference on Business Process Management, France, pp. 302-318.

BEA Corporation. (2005). “Domain Model for SOA, Realizing the Business Benefit of Service-Oriented Architecture,” http://files.shareholder.com/downloads/BEAS.

Boehm, B., R. Valerdi, J. Lane, and W. Brown. (2005). “ COCOMO Suite Methodology and Evolution, ”

http://sunset.usc.edu/publications/TECHRPTS/2005/usccse2005-509/usccse2005-509.pdf.

Brooks, F. P., Jr. (1996). “The Computer Scientist as Toolsmith II,” *Communications of the ACM*, (39)3, pp 61-68.

Brown, A. W., M. Delbaere, P. Eeles, S. Johnston, and R. Weaver. (2005). “Realizing Service-Oriented Solutions with the IBM Rational Software Development Platform,” *IBM Systems Journal*, (44)4, pp. 727-752.

Brown, J. S., A. Collins, and S. Duguid. (1989). "Situated Cognition and the Culture of Learning," *Educational Researcher*, vol. 18, pp. 32-42.

Brownsword, L., D. Fisher, E. Morris, J. Smith, and P. Kirwan. (2006). “ System-of-Systems Navigator: An Approach for Managing System-of-Systems Interoperability, ” in Technical Note CMU/SEI-2006-TN-019.

Cargill, C. (1995). “A Five-Segment Model of Standardization.” in Kahin and Ahhate (eds.), *Standards Policy for Information Infrastructure*, Cambridge: MIT Press.

Cargill, C. (1995). *Information Technology Standardization: Theory, Process, and Organizations*, Digital Press, Newton, MA, USA.

Charette, R. N. (2005). “ Why Software Fails, ” IEEE Spectrum. http://www.spectrum.ieee.org/sep05/1685.

Checkland, P. and J. Scholes. (1990). *Soft Systems Methodology in Practice*, J. Wiley.

CIO Magazine. (2007). ABC: An Introduction to Service-oriented Architecture (SOA), http://www.cio.com/article/40941.

CRA. (2006). “ Interest in CS and CE as Majors Drops in 2005, ” in CRA Bulletin, http://www.cra.org/wp/index.php?p=75.

Crawford, C. H., G. P. Bate, L. Cherbakov, K. Holley, and C. Tsocanos. (2005). “Toward an On-Demand Service-Oriented Architecture,” *IBM Systems Journal* (44)1, pp. 81-107.

Edens, K. M. (2000). “ Preparing Problem Solvers for the 21st Century through Problem-Based Learning, ” College Teaching vol. 48, pp. 55-60.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology and Design*, Prentice-Hall.

Fomin, V., T. Keil, and K. Lyytinen. (2003). “Theorizing about Standardization: Integrating Fragments of Process Theory in Light of Telecommunication Standardization Wars,” Sprouts: Working Papers on Information Environments Systems and Organizations.

Gartner, Inc. (2005). “ Benefits and Challenges of SOA in Business Terms, ” 6 September 2005.

Ghemawat, Pankaj. (2007). “Managing Differences: The Central Challenge of Global Strategy,” *Harvard Business School, Published on Mar 1, 2007*.

Goul, M. (2007). “What’s Your Strategy for Master Data Management?” Teradata White Paper, http://www.teradata.com/t/page/166150/index.html?src=tdmo_rl&i=v07n03.

Honey, Alan. (2007). “SOA and HL7 V3 Proposal Overview,” *Kaiser Permanente Presentation.*

Horn, Paul. (2005). “The New Discipline of Services Science,” *BusinessWeek*, http://www.businessweek.com/technology/content/jan2005/tc20050121_8020.htm.

HR1086. (2003). “Standards Development Organization Advancement Act of 2003,” the Senate and House of Representatives of the United States of America.

Huhns, M. N. and M. P. Singh. (2005). “Service-Oriented Computing: Key Concepts and Principles,” *IEEE Internet Computing, (9)1, pp. 75-81*.

IBM Academic Initiative. (2007). “Service-Oriented Architecture Courses,” http://www-304.ibm.com/jct09002c/university/scholars/courseware/soa.html.

InformationWeek. (2007). InformationWeek 500 Research Report, http://www.informationweek.com/iw500.

Jain, H. and P. Vitharana (eds.) (2008) “Special Issue on Architecture & Design for Application Agility,” Information Technology and Management (9)1, pp. 1-3.

Keil, M. and D. Robey. (1999). “Turning around Troubled Software Projects: An Exploratory Study of the De-Escalation of Commitment to Failing Courses of Action,” *Journal of Management Information Systems*, 15, pp. 63-87.

Kirsch, L. J. (1996). “The Management of Complex Tasks in Organizations: Controlling the Systems Development Process,” *Organization Science* 7, pp. 1-21.

Kirsch, L. J. (2002). “Controlling Information Systems Development Projects: The View from the Client”, *Management Science* 48, pp. 484-498.

Lang, K. R. and J. L. Zhao. (2000). “ The Role of Electronic Commerce in the Transformation of Distance Education, ” *Journal of Organizational Computing and Electronic Commerce*, (10)2, pp.103-28.

Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-Network-Theory*, Oxford: Oxford University Press.

Leymann, F., D. Roller, and M. T. Schmidt. (2002). “Web Services and Business Process Management,” *IBM Systems Journal* (41)2, pp. 198-226.

McCallum, S. H. (1994). “Information Technology Standards: Implementation, Maintenance, and Coordination,” In William E. Moen (ed.), *Scholarly Information and Standardization* (pp. 13-20), Bethesda, MD: NISO Press.

Mitra, P., S. Purao, J. W. Bagby, K. Umapathy, and S. A. Paul. (2005). “An Empirical Analysis of Development Processes for Anticipatory Standards,” NET Institute Working Paper No. 05-18. Available at SSRN: http://ssrn.com/abstract=850524.

Moen, W. (1998). *The development of ANSI/NISO Z39.50: A Case Study in Standards Evolution*, School of Information Studies, Syracuse University.

Nilsen, H. and S. Purao. (2005). “ Balancing Objectivist and Constructivist Pedagogies for Teaching Emerging Technologies: Evidence from a Scandinavian Case Study, ” *Journal of Information Systems Education* 16, pp. 281-293.

OMB. (1998). OMB Circular A 119, Federal Participation in the Development and Use of Voluntary Consensus Standards and in Conformity Assessment Activities, http://www.whitehouse.gov/omb/circulars/a119/a119.html.

Peltz, C. (2003). “ Web Services Orchestration: A Review of Emerging Technologies, Tools, and Standards, ” Hewlett Packard Labs Technical Paper.

Schmidt, J. (2004). A Methodology for Enterprise Integration, http://www.wwintegration.com/eai_methodology.htm.

Simon, H. A. (1976). *The Sciences of the Artificial*, Cambridge, MA: MIT Press.

Singh, M. P. and M. N. Huhns. (2005). *Service-Oriented Computing*, John Wiley.

Snow, A. and M. Keil. (2002). “The Challenge of Accurate Software Project Status Reporting: A Two-Stage Model Incorporating Status Errors and Reporting Bias,” *IEEE Trans. on Engineering Management* 49, pp. 491-504.

SOA Consortium. (2007). SOA Executive Insight Report, http://www.soa-consortium.org/Executive_Insight_whitepaper-2007.pdf.

SOAP. (2003). Simple Object Access Protocol (SOAP) Version 1.2, http://www.w3.org/TR/soap12-part1.

Tan, B. C. Y., H. J. Smith, and R. Montealegre. (2003). “Reporting Bad News about Software Projects: Impact of Organizational Climate and Information Asymmetry in an Individualistic and a Collectivistic Culture,” *IEEE Transactions on Engineering Management* 50, pp. 64-77.

UDDI. (2005). Universal Description Discovery and Integration (UDDI), http://uddi.org/pubs/uddi-v3.0.2-20041019.htm.

Umapathy, K., and S. Purao. (2007). “A Theoretical Investigation of Emerging Standards for Web Services,” *Information Systems Frontiers Special Issue on From Web Services to Services Computing: Technologies and Applications* (9)1, pp. 119-134.

Umapathy, K., S. Purao, and V. Sugumaran. (2003). “ Facilitating Conversations among Web Services as Speech-act based Discourses, ” Proceedings of the Workshop on Information Technologies and Systems (WITS), Seattle, WA.

Virili, F. (2003). “Design, Sense-Making and Negotiation Activities in the ‘Web Services‘ Standardization Process,” *MISQ* Special Issue Workshop on Standards Making.

Vitharana, P., K. Bhaskaran, H. Jain, H. J. Wang, and J. L. Zhao. (2007). “Service-Oriented Enterprises and Architectures: State of the Art and Research Opportunities,” Proceedings of the Americas Conference on Information Systems, Keystone, CO, U.S.A.

Vitharana, P., H. Jain, and F. M. Zahedi. (2004). “Strategy-Based Design of Reusable Business Components,” *IEEE Transactions on Systems, Man, and Cybernetics* (34)4, pp. 460-474.

Watts, H. (2006). *Systems of Systems: Scaling up the Development Process*, Software Engineering Institute, Carnegie Mellon University.

Weick, K. (1995). *Sensemaking in organizations*, Sage Publications.

WSDL. (2001). Web Service Description Lanaguage (WSDL), http://www.w3.org/TR/wsdl.

Wysocki, B. (1998). “ Pulling the Plug: Some Firms, Let Down by Costly Computers, Opt to ‘De-Engineer’, ” in *The Wall Street Journal*.

Zimmermann, O., M. R. Tomlinson, and S. Peuser. (2005). *Perspectives on Web Services: Applying SOAP, WSDL, and UDDI to Real-World Projects*, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Zhao, J. L., M. Tanniru, and L. Zhang. (2007). “Service Computing as the Foundation for Enterprise Agility: Overview of Recent Advances and Introduction to the Special Issue,” *Information Systems Frontiers,* (9)1, pp. 1-8.

ABOUT THE AUTHORS

J. Leon Zhao is Eller Professor and Interim Head of the Department of MIS, University of Arizona. He taught previously at HKUST and College of William and Mary, respectively. He holds Ph.D. and M.S. from Haas School of Business, UC Berkeley, M.S. from UC Davis, and B.S. from Beijing Institute of Agricultural Mechanization. He has published more than 120 research articles in major journal and conference outlets. He received an IBM Faculty Award in 2005 for his work in business process management and services computing. Leon has been associate editor of Information Systems Research and Decision Support Systems among others. He has co-edited nine special issues in various IS journals and has chaired numerous academic conferences including the Workshop on E-Business (2003), the 15th Workshop on Information Technology and Systems (2005), the IEEE Conference on Services Computing (2006), and the First China Summer Workshop on Information Management (2007), the IEEE International Symposium on Advanced Management of Information for Globalized Enterprises (2008), and the Arizona Exposium on Frontiers in Information Technology and Applications (2008).

Michael Goul is Professor in the W. P. Carey School of Business at Arizona State University. His interests in services-centric computing have been motivated by action research projects with companies including Teradata, American Express and Intel. His interests also extend to the public sector as evidenced by his appointment as a Distinguished Fellow of the William J. Clinton School of Public Services. Dr. Goul has authored or co-authored over fifty journal and refereed proceedings papers, he has supervised sixteen doctoral dissertations and over sixty master degree theses. He has served as a journal editor, special issue co-editor, Association for Information Systems Vice-President, Conference Chair, Program Chair and Special Interest Group in Decision Support Chair and Vice-Chair. He has received several graduate faculty teaching awards and he has taught numerous classes across the W.P. Carey School's undergraduate, MBA, MS and Ph.D. platforms.

Sandeep Purao is Associate Professor at the College of Information Sciences and Technology and part of the Enterprise Informatics and Integration Center at Penn State University. His research deals with design, evolution and management of complex techno-organizational systems blending research methods from social science and software engineering. His research has resulted in publications in archival journals such as CACM, ISR, JMIS, DSS, ACM Computing Surveys; papers in conferences such as ICIS and WITS; and the creation of software artifacts such as APSARA. He holds a Ph.D. in Management Information Systems from the University of Wisconsin-Milwaukee. He currently serves as the Graduate Student Adviser at the College of IST; and teaches organizational informatics and enterprise integration.

Padmal Vitharana is Associate Professor of information systems in the Whitman School of Management at Syracuse University. He received his Bachelor of Science, Master of Business Administration and Doctor of Philosophy degrees from the University of Wisconsin system. His research interests lie in software development in general and reusable component/services in particular. His research has been published in journals such as the *Communications of the ACM, IEEE Transactions on Software Engineering, IEEE Transactions on Man, Systems, and Cybernetics, Information and Management,* and *Database for Advances in Information Systems.* He also edited a special issue on service-based development titled “Architecture and Design for Application Agility” in Information Technology and Management journal.

Harry Jiannan Wang is Assistant Professor of Management Information Systems in the Lerner College of Business and Economics, University of Delaware. He received a Ph.D. in MIS from Eller College of Management, University of Arizona and a B.S. in MIS from Tianjin University, China. His research interests involve business process management, workflow technologies and applications, service oriented architectures, Web services, and access control in e-commerce.

Communications of the Association for Information Systems

ISSN: 1529-3181