Journal of Internet Commerce
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/wico20

Using Web Services for Collaborative, Inter-Organizational Computing: A Framework and Implementation

Ananth Srinivasan PhD [a] & David Sundaram PhD [b]

[a] Department of Information Systems and Operations Management, University of Auckland, Auckland, New Zealand E-mail: a.srinivasan@auckland.ac.nz

[b] Department of Information Systems and Operations Management, University of Auckland, Auckland, New Zealand E-mail: d.sundaram@auckland.ac.nz

Available online: 05 Nov 2008

To cite this article: Ananth Srinivasan PhD & David Sundaram PhD (2006): Using Web Services for Collaborative, Inter-Organizational Computing: A Framework and Implementation, Journal of Internet Commerce, 5:4, 5-29

To link to this article: http://dx.doi.org/10.1300/J179v05n04_02

Using Web Services for Collaborative, Inter-Organizational Computing: A Framework and Implementation

Ananth Srinivasan
David Sundaram

ABSTRACT. Collaborative computing is a powerful approach to facilitate service delivery across organizational boundaries. The major drawback of many of the current approaches of dealing with this issue is the dependence on proprietary solutions that are not based on open standards. Such solutions hinder the smooth exchange among the participating organizations due to their complexity and lack of interoperability. Web Services (WS) offer the promise of streamlining the delivery of e-services across organizational boundaries. What is currently missing is (1) a clear understanding of the important issues in deploying these technologies and (2) a compelling implementation framework for the deployment of these technologies in organizations. In this research we propose a conceptual framework for building systems that utilize Web Services technologies to enable networked organizations to automate collaborative processes. We propose an e-Business Service Chain grid that provides an impetus for addressing some of the gaps that exist in current e-services business and delivery models. doi:10.1300/J179v05n04_02

Ananth Srinivasan, PhD (E-mail: a.srinivasan@auckland.ac.nz), is Professor of Information Systems and Digital Commerce and Co-Director of the Centre for Digital Enterprise and David Sundaram, PhD (E-mail: d.sundaram@auckland.ac.nz), is Senior Lecturer, both at Department of Information Systems and Operations Management University of Auckland, Auckland, New Zealand.

The authors would like to acknowledge the contributions of Schalk Pienaar, Nikhil Ravishankar, and McNeill Mendes toward different aspects of the project.

Journal of Internet Commerce, Vol. 5(4) 2006
Available online at http://jicom.haworthpress.com

doi:10.1300/J179v05n04_02

KEYWORDS: e-Services delivery, Web Services orchestration, inter-organizational systems, enterprise collaboration

INTRODUCTION AND PROBLEM STATEMENT

Internet technologies are widely recognized for their promise as enablers of collaborative computing both within and among organizations. The presence of heterogeneous systems based on different technological platforms in organizations makes the implementation of network collaboration across organizational boundaries very complex (Stal 2002). The approach taken for the most part to deal with this issue has been based on Enterprise Application Integration (McKeen and Smith 2002). The major drawback of this approach is the dependence on proprietary solutions that are not based on open standards (Fremantle, Weerawarana, and Khalaf 2002). When the need for inter-organizational collaboration arises, such solutions hinder the smooth exchange among the participating organizations due to their complexity and lack of interoperability. "Web Services" is a new class of internet based, open standards technology that offers the promise of resolving these problems (Gottschalk 2000; IBM 2003). What is currently missing is a compelling implementation framework for the deployment of this technology in organizations. A lack of clear understanding about how to deploy Web Services to enable inter-organizational collaboration will impede the uptake of this promising new technology. In this project, our aim is to construct and test a valid implementation framework for Web Services. Such a framework will enable effective inter-organizational network-based computing which will have a positive effect on organizational productivity.

A reality of modern organizations and the manner in which they function is one of collaborative work. Perhaps the two most important business models that exemplify collaborative work that is largely enabled by a technological infrastructure are supply chains and e-business. In both cases, it is imperative that business processes that span organizational boundaries be defined and supported for the efficient functioning of the participants. These processes may span multiple

boundaries in order to completely define a particular unit of work in a focal organization. The infrastructure that supports such processes must take into account the fact that multiple organizations must work in a synchronized manner to accomplish organizational goals. The requirements under such a scenario now go one level beyond looking at a single enterprise to accounting for multiple enterprises working in a coordinated fashion. High-level modeling now becomes a core aspect defining the requirements in a way that ensures that the systems that are designed to support such activity align properly with multi-enterprise level objectives. Requirements that are determined in this fashion can ensure that systems work efficiently and in unison to accomplish the goals.

Integration of systems that exist in multiple organizations is the key to success in a multi-organizational environment. The issue of IT relatedness across multiple business units has been highlighted as a significant antecedent to organizational performance (Stal 2002). However, achieving proper integration is often fraught with risks due to poorly defined business processes and technological heterogeneity. In this paper, we outline a framework to overcome such risks by explicitly modeling business processes. We consider variations in both business process complexity and organizational boundary definitions and the implications for design. Finally, by example, we show a design prototype that demonstrates what the framework is trying to achieve by using Web Services technologies to implement the requirements identified by high-level modeling.

WEB SERVICES: THE ESSENTIALS

Web Services promise cheap, reliable, flexible, and scalable computing. Further, their deployment and maintenance is not very complex. Though much hype still surrounds them, most practitioners agree that they offer a significant step forward in leveraging the power of the Internet. Web Services have received wide-spread support from industry leaders. These organizations have been instrumental in driving the setting of standards and specifications for the development of Web Services and their related technologies. This task is overseen currently by the *W3C Working Groups* (http://www.w3c.org/2002/ws). Another possible reason for this enthusiasm is summarized well by Johnston (2002), who suggests that while integrating technologies already exist, limitations such as complexity and verbosity have impacted their widespread use. Web Services address these issues by steering a middle road between complexity and verbosity.

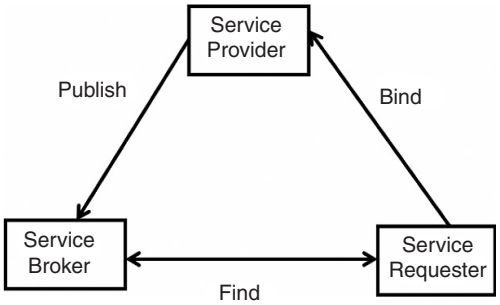
A brief definition of Web Services highlights the following: They are applications (operations and tasks) that can be developed in modular, independent fashion and made available through their interfaces across heterogeneous systems using standard internet protocols (Dustdar 2004; Gottschalk 2000). A Web Service is less a single tangible thing than an interwoven mesh of technologies and standards that work together to achieve their shared goals. More specifically Web Services are available over the internet and they communicate using the eXtensible Markup Language (XML). All communications "to" and "from" a Web Service are encoded using XML. Clients invoke a Web Service by sending an XML request/message and await for a response from the Web Service which is also encoded in XML.

A Web Service architecture requires three fundamental operations: publish, find, and bind (Gottschalk 2000) (Figure 1). Service providers publish services to a broker while service requesters find required services using a broker and bind to them. Each of these fundamental operations is enabled through one or more of the several technologies and standards that we encompass in the term "Web Services." At the heart of Web Services lies XML. The *Web Service Description Language* (WSDL), and the *Simple Object Access Protocol* (SOAP) are two XML-based technologies that Web Services use. WSDL is used to publish a Web Service and act as an instruction manual for binding to it. SOAP is used to encode and transfer messages to and from Web Services.

SERVICE CHAIN ORCHESTRATION

Fremantle et al. (2002) proposed the idea of a service oriented vision for enterprise architecture, where components described in *Web Servi-*

FIGURE 1. Publish, Find, and Bind (Adapted from Gottschalk, 2000)



ces Definition Language (WSDL) are registered and called upon by other components when they are required. In the service oriented vision, components are choreographed to form a chain of components, performing various enterprise tasks. We call these "integrated components service chains."

The service-oriented vision is driven largely by two themes: (1) the first is e-Business, in which organizations are increasingly exposing and integrating their processes, and (2) the second is business process automation, which is the systematic integration of everyday business processes by integrating core systems. The service-oriented vision has many benefits including the ability to expose existing functionality through WSDL, moderating heterogeneity, providing interoperability, and supporting business processes.

Middleware applications that enable business workflow, such as *Microsoft's Biztalk*, already exist. However, Web Service composition (the sequencing of Web Services into a business process or workflow) faces many challenges for it to be coordinated consistently independent of the middleware. Dustdar (2004) describes the main problems as follows:

- How can abstract Web Services Workflows be represented (e.g., visually), and be executed in the form described?
- How can Web Services be discovered and the interoperability guaranteed?
- How can one ensure efficiency of composed Web Services?

Two key initiatives that support this vision are the *Web Services Choreography Interface* (Arkin, Askary, Fordin, Jekeli, Kawaguchi, Orchard, Pogliani, Riemer, Struble, Takacsi-Nagy, Trickovic, and Zimek 2002) and the *Business Process Execution Language for Web Services* (Curbera, Goland, Klein, Leymann, Roller, Thatte, and Weerawarana 2002). The *Web Services Choreography Interface* (WSCI) initiative is spearheaded by *BEA Systems, Intalio, SAP,* and *Sun Microsystems*. It is an XML-based language that describes and provides a global picture of the messages exchanged by interacting Web Services. The *Business Process Execution Language for Web Services* (BPEL4WS) is spearheaded by *IBM* and *Microsoft*. It combines *IBM*'s *Web Services Flow Language* (WSFL) technologies and *Microsoft's Xlang* and defines a notation for describing the behaviour of business processes that are based on Web Services.

One common application of e-Business is inter-organizational value chain building, where a business process spans across systems of multiple organizations, adding value at each node. While current architectures like CORBA and COM+ perform these tasks well, they are tightly coupled and expensive. Web Services, while less sophisticated than these technologies, are natural candidates to replace them because of the simplicity and modularity they offer.

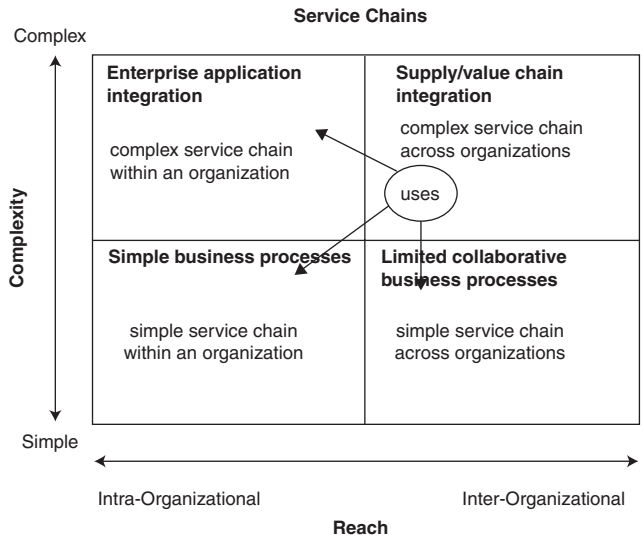
A construct that outlines how service chains can support e-Business and the service-oriented vision is the service chain complexity grid (Figure 2). The grid focuses on the dimensions of complexity and reach of chains of Web Services. We have singled out these dimensions because we believe they are the critical enablers of e-Business. However, many of these dimensions are inter-related in particular ways.

For example, if a Web Service is dynamic rather than static, nested or composite rather than flat, or versatile rather than specific, it is also more likely to be complex, rather than simple. Essentially, if a Web Service performs a simple task, it is probably a fairly simple Web Service. Conversely, complex tasks are performed by more complex Web Services. The reach of a Web Service is another critical measure of its usefulness in e-Business activities. Reach refers to the Web Service's proclivity to communicate across numerous distinct systems that may belong to a single organization or span organizational boundaries.

All e-Business activities typically would fall somewhere in the quadrants of this grid (Figure 2). From operational activities and processes at the bottom left of the grid, to more strategic processes at the top right. The first quadrant (bottom left hand) describes simple business processes. These processes are the building blocks of e-Business, used to perform simple operational tasks. Quadrant two (on the bottom right hand) of the grid represents business processes which are simplistic but beginning to show a collaborative theme as they span organizational boundaries. e-Business activities are not always conducted across organizational boundaries, however. Often there are intra-organizational processes which leverage e-Business models. Usually these cases involve inter-departmental processes or communication across internal system boundaries. Enterprise application integration is an example of this type of activity, which is found in quadrant three (top left hand). Quadrant four of the grid is the most sophisticated area of activity, with complex inter-organizational processes being employed to provide strategic collaboration between business partners.

Dynamic collaborations between business processes often require peer-to-peer interaction. Figure 3 illustrates the peer-to-peer service

FIGURE 2. The e-Business Service Chain Complexity Grid

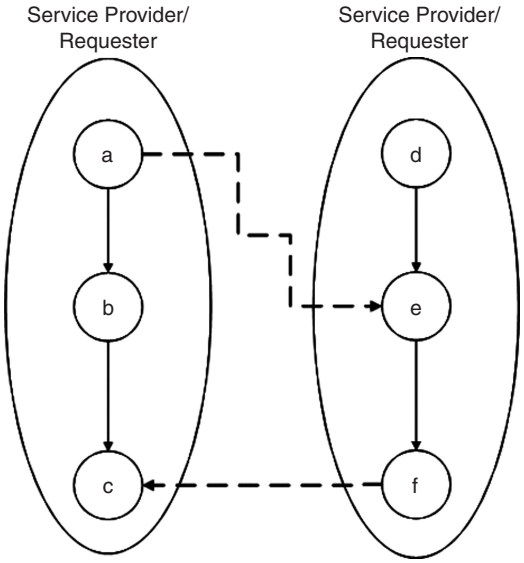


model, and is a good example of the collaborations found in the supply/value chain integration quadrant of the grid. Here we see an example of the interaction between two business partners and their processes. Service **a** of the first process requires service **e** of the second process and service **f** of the second process requires service **c** of the first process. The two business partners in this scenario switch between the roles of the service provider and service requestor. The two business process flows communicate with each other to achieve a single purpose. While this example just shows two interacting processes between two business partners the concept can be extended to many other processes and/or partners.

BUSINESS PROCESSES AND WORKFLOW

The term business process is used and defined with varying meanings. Davenport and Short (1990) define business process as "a set of logically related tasks performed to achieve a defined business outcome." A business process can be implemented either as a material pro-

FIGURE 3. Peer-to-Peer Interaction of Two Business Processes (Adapted from Leymann et al., 2002)



cess where the outcome is a physical manifestation or as an information process where the outcome is that of providing a service. The latter category of business process that focuses on delivering a service is often referred to as workflow (Reijers 2003). Another popular interpretation of workflow is quite simply the automation of business processes. The *Workflow Management Coalition* (WfMC) defines workflow as: "The automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules" (Lawrence 1997). They go on to describe *Workflow Management Systems* (WfMS) as: "A system that defines, creates and manages the execution of workflows through the use of software, running one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications" (Lawrence 1997). A WfMS needs to support at least four of the basic coordination constructs or patterns: sequential routing, conditional routing, parallel routing, and iterative routing. Several other more complex patterns have been identified as being necessary to model real-world workflows (van der Aalst and Kumar 2001). One of the major drawbacks of traditional WfMS such as *COSA, HP Changengine,*

Fort, Conductor, I-Flow, InConcert, MQ Series Workflow, R/3 Workflow, Staffware, Verve, and *Visual WorkFlo* is their lack of interoperability and their use of proprietary formats to represent workflow schemes (van der Aalst and Kumar 2001).

WEB SERVICES APPLICATION CASE STUDIES

To illustrate important dimensions of the use of Web Services, we describe the deployment of the technology in a variety of organizational settings. These examples help us better understand the full range of issues that arise in the use of the technology and the specific organization processes that are impacted by them. They also help us begin to articulate a sensible framework that captures the various dimensions in a manner that fosters systematic development of ideas for web service implementation.

Sun Microsystems, Inc.

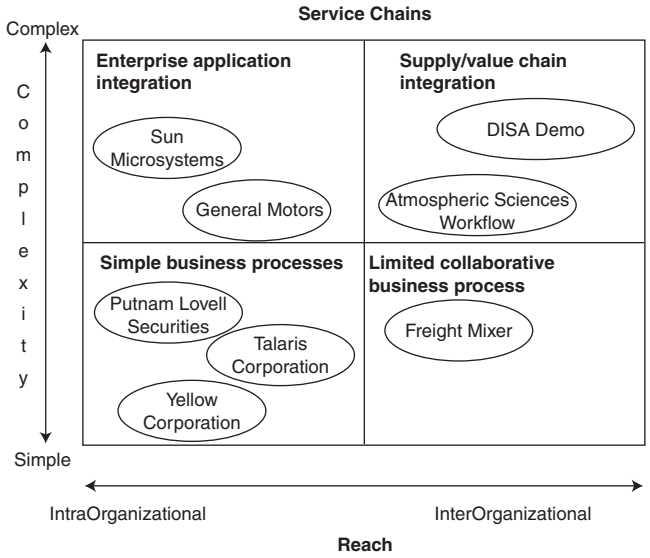
With over 80 Web Services in production or development, the designers at *Sun Microsystems* have designed services which are used by employees, suppliers, partners, and customers (Aldrich 2003). Here we look at *e-Payment*, an application that was configured to provide a credit card processing facility. It is made up of other Web Services that support financial transactions, including a currency converter and a zip code verifier, and it also shares a set of core Web Services for authentication, authorization, management, monitoring, measurement, and reliable messaging. This makes the *e-Payment* application a loosely coupled *ad hoc* type of application. The application does not cross organisational boundaries and is an intra-organisational web service implementation. This application, made up of several Web Services of varying complexities, can be classified as a complex web service, based on the cost and complexity *Sun* faced for developing individual versions of this capability in several previous applications. This application does not just provide a pre-defined set of data but provides an entire credit card processing service with results that vary from transaction to transaction. However, the results themselves are part of a pre-defined set of options and therefore this application can be classified as a static Web Service application. Finally, the *e-Payment* application, which is made up of several smaller Web Services and uses functionality of other Web Services, can be classified as a nested Web Service application which

provides both new functionality as well as wrappers to already existing functionality. Wrappers have traditionally been used to wrap legacy applications and make them usable in an object-oriented/component-based environment. But wrappers are increasingly being used to wrap legacy and monolithic software into Web Services and/or Web Services-accessible applications. Based on our analysis the implementation can be classed in the Enterprise Application Integration quadrant of the e-business service chain grid (Figure 4). The application is based on a number of core Web Services that could be used in various other applications. This implementation can be described as moderately versatile (as it can be used in multiple application domains) and moderately usable (as it is fairly simple to embed this service as part of other applications).

FreightMixer

The *FreightMixer* application (Piccinelli and Williams 2003) was designed to offer an end-to-end freight service without owning fixed transport or transport-related infrastructure. *FreightMixer* is used to deliver packaged solution to customers by combining services of other

FIGURE 4. Integration e-Business Chain Grid



freight service providers. Using this application service, providers can interact directly with customers or with each other based on the supervision of and composition logic decided by *FreightMixer*. These direct interactions are completely transparent to the customer.

In *FreightMixer,* various services are orchestrated together in an *ad hoc* fashion to achieve the customers' needs. The entire workflow of providing freight services spans several organizations and in fact is based solely on services provided by external organizations, with *FreightMixer* acting as an intermediary between these organizations and the customer. This web service-based workflow implementation can be classed as an inter-organizational application. Here the individual services themselves are simple; however the workflow aspects of service execution and creation of the package for the customers are complex tasks. *FreightMixer* is a completely dynamic implementation and provides the most optimum solution based on the customers' needs. The application in itself is a Web Service made up of several participating Web Services and can be described as a nested workflow implementation providing new functionality. As a moderately complex inter-organizational workflow implementation *FreightMixer* can be classed high up in the Limited Collaboration Business Processes quadrant of the e-business service chain grid (Figure 4). The implementation is domain specific but is versatile enough to solve a wide range of freight service requests using the *DySCo* framework, but the complexity is fairly well hidden making it quite user-friendly. This implementation can be placed in the centre of the graph and can be described as moderately versatile and usable.

Yellow Corporation

Yellow Corporation offers a wide range of shipping services for moving industrial, commercial, and retail goods. Three new Web Service applications to track shipments, automatically create purchase orders, and get rate quotes, were developed. In all three cases Web Services were front-end wrappers to existing back-end legacy systems. Gralla (2003) identifies this as one of the key issues as no changes needed to be done to *Yellow*'s backend systems to provide these new services. These three Web Service applications can be accessed through Internet based applications for individual users, or hand-held devices, and they can also be directly included in other applications enabling *Yellow* customers to include them in their own business's workflow.

Unlike the previous *FreightMixer* case study, the Web Services are tightly coupled and can therefore be referred to as "negotiated Web Ser-

vices." They are also intra-organizational; however, they can become part of inter-organizational workflows if integrated by their customers as part of their own business processes. These web service applications are simple in functionality but dynamic in the nature of their output. Though the domain is the same, these Web Services are simpler in comparison to the *FreightMixer* example but still deliver substantial benefits to *Yellow Corporation*. This implementation can be classed ias the Simple Business Process quadrant of the e-business service chain grid (Figure 4). This implementation can be described as highly usable but not versatile as it supports just one of type of application and operates in a single application domain.

Atmospheric Sciences Workflow

This Web Service implementation is based on solving atmospheric sciences problems using the concept of grid workflows. (Abramson, Komineni, McGregor, and Katzfey 2004). Grid workflows consist of various components such as computational models, distributed files, scientific instruments, and specialized hardware platforms. These components are dynamically linked together to solve complex problems,. The Grid workflow system couples existing legacy system models, such as Global Climate Models and Regional Climate Models, in a distributed environment using Web Services.

This atmospheric sciences web service based workflow implementation is loosely coupled and can be classed as an *Ad-Hoc* Web Service implementation. It spans multiple organizations and acts as wrappers to very complex atmospheric models. However, new functionality has also been implemented using Web Service technology. As far as the workflow orchestration is concerned the application is nested and can be configured dynamically based on the nature of the atmospheric sciences problem being solved. Of the four case studies discussed so far this implementation is the most complex and can be classed ias the Supply Chain Integration quadrant of the e-business service chain grid (Figure 4). Though the implementation is domain-specific, it is versatile enough to solve a wide range of climate prediction problems using the Grid workflow framework.

Defence Information Systems Agency (DISA)

Due to recent terrorist events in the United States and around the world, DISA were faced with the challenge of handling such threats.

They identified the need to assimilate, analyse, and redistribute information quickly and efficiently using new technologies. DISA identified that the way to manage or prevent crisies was by getting information to the right agencies at the right time. A demonstration was conducted by DISA (2005) along with IBM in December 2002 to address the issue of technology and homeland security. *Homeland Security Command and Control Advance Concept Technology Demonstration* or HLSC2 ACTD was designed to simulate a multiple day scenario where a number of attacks were made at different locations around the country. Web Services technologies were used to enable communication and coordination between the different components within the Federal, State, local Government, and Civil authorities (IBM 2003).

Depending on the type of the incident or crisis, the implementation was designed to dynamically link several different services to manage or respond to that incident. Therefore HLSC2 ACTD is a loosely coupled *ad hoc* application. With its primary goal being the efficient deployment of information between the various agencies of the DOD it is a complex inter-organizational workflow implementation. This application dynamically provides different responses to different situations. The Web Services were designed to act as interfaces or wrappers to existing functionality. On the e-business service chain grid (Figure 4) this application can be placed in the Supply Chain Integration quadrant, as it is a highly complex inter-organizational workflow implementation. While the application is extremely versatile in the various scenarios it can handle, it is not very user-friendly and only specialists can use it with ease.

General Motors

General Motors (GM) spends about $ 4 billion a year on acquiring new software (Business Week Online, June 24, 2003). In an effort to curtail expenditure on the acquisition of new software, GM is deploying Web Services to upgrade existing software and integrate applications across its many systems. The technology is being used to act as a bridge between a diverse set of systems within the organization. With increasing amounts of software placed in vehicles themselves, the objective in the future is to use Web Services to upgrade software that is resident in these vehicles. The technology can help cut the cost associated with upgrading such software. Another useful application that GM sees for the technology is to connect its parts and inventory data in flexible ways with its suppliers. Once issues related to security are resolved satisfac-

torily, the push will be toward tying together data from a diverse set of systems that span organizational boundaries. But at the moment the Web Services are intra-organizational and of reasonable complexity. Hence we place the GM services in the Enterprise Application quadrant of the e-business service chain grid (Figure 4).

Putnam Lovell Securities

Putnam Lovell Securities, an investment bank, uses Web Services to automate the process of customizing the content of their research department to suit the particular needs of their client base (Nghiem 2002). The emphasis on the use of Web Services was on integrating a variety of applications. Both, information about customers and research content of interest to their clients, are applications that reside outside the organization and are accessible through the internet infrastructure. The leverage gained by Web Services technologies is the automation of the collection of research content distributed by multiple sources and the matching of the content to the specific requirements of the investment clientele. Real value is added by conforming to the specific formatting requirements of various clients whether they are investors or distributors of research content. For example, individual clients can receive tailored information by e-mail; a distributor can get the same information in a portable document format. Web Services has helped the organization reduce the costs, increased the level of personalization, and improved the speed with which information gathered from multiple sources external to the organization are distributed to clients. Most of the services are defined for integrating multiple applications, though the sources and consumers are external, hence we consider the Web Services to be primarily intra organizational. The level of complexity is deemed to be low since it is primarily mapping of diverse sources that is being carried out. Hence we place the *Putnam Lovell Securities* services in the Simple Business Process quadrant of the e-business service chain grid (Figure 4).

Talaris Corporation

Talaris Corporation (Gralla 2003) is an excellent example of the use of Web Services technologies to deploy services-based procurement platforms for client organizations. Web Services technologies are used to maintain a private registry of service providers and detailed profiles of individual clients. When a client organization requires a business ser-

vice such as a meeting facility or an airline reservation, the registry is processed in a way that matches the specific requirements of the client organization and a solution is offered. The strengths of the application in *Talaris* lie in (a1) compliance with specific client organization procurement policies, (b2) knowledge of individual client profiles to produce a personalized solution, and (c3) offering services from suppliers who are approved by client organizations. Since the Web Services are for internal consumption and of medium level complexity we place the *Talaris Corporation*'s services in the Simple Business Process quadrant of the e-business service chain grid (Figure 4).

One important aspect to note about the above case studies is that the only case studies which fit into the Supply Chain integration quadrant, the Atmospheric Sciences and DISA implementations, are either research prototypes or demonstrations. There are many common threads that run through the various applications described above. First, is the ease of integration that is offered by Web Services technologies. The use of open, non-proprietary standards that can be deployed over the Internet is perhaps the prime catalyst that encourages its use. Second, the ability to integrate diverse systems regardless of whether they are within a single organization or span organizational boundaries addresses a problem that has plagued organizations for many decades. Third, the ability to provide personalized solutions to information needs based on a deep understanding of client profiles makes it possible to accurately align specific business processes with deployment of the technology.

WORKFLOW-BASED WEB SERVICES ORCHESTRATION

A system that orchestrates diverse Web Services over a distributed environment needs to be able to find them, and to bind to them together somehow in a meaningful fashion. The left-hand side of Figure 3 illustrates the broad vision of what such a system would achieve.

At the core of the system is a kernel that supports the plugging-in of any combination of services to create a service chain which supports the execution of a business process scenario. The *scenario* is intended to allow for the construction of a useful business process on an *ad-hoc* basis. Stable and well recognized business processes exist in organizations and these have been the focus of much of the existing work. The combination of such stable stability with *ad-hoc* processes needs to be explicitly acknowledged in a true service-oriented framework. We

think of a scenario in this case as an instance of a process. Services may come from within an organization, another organization, or independent service providers. These providers could either be service development specialists or other application vendors or developers exposing useful functionality of existing systems.

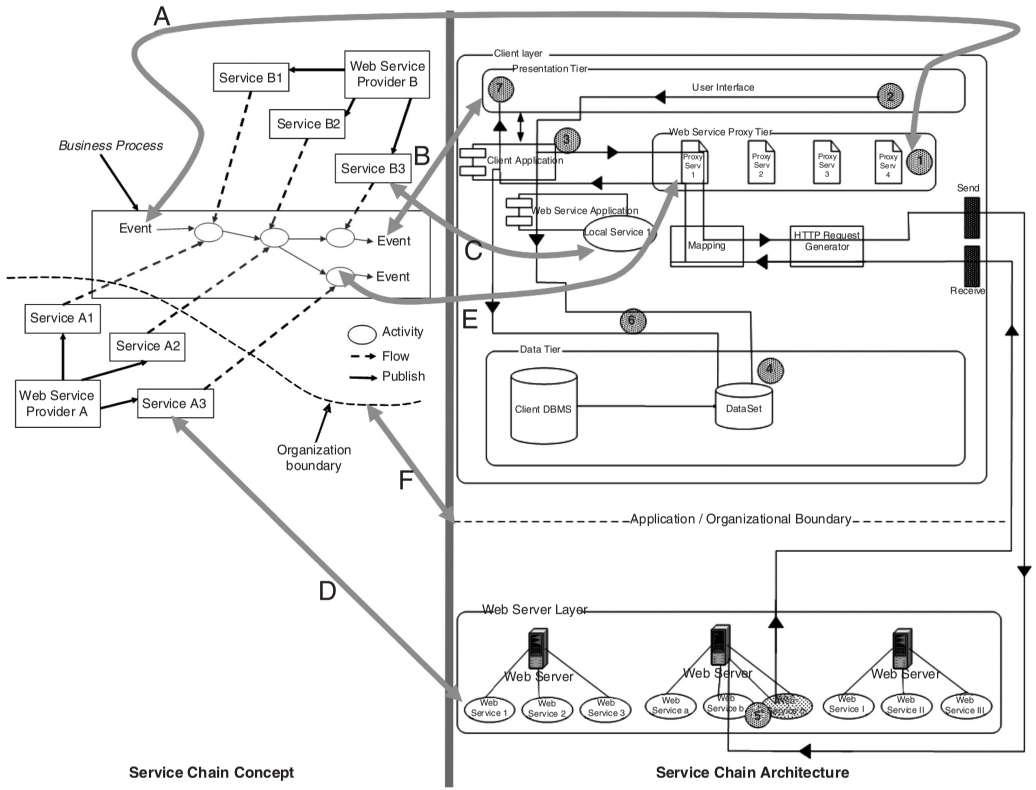
A SPECIFIC APPLICATION THAT INTEGRATES MULTIPLE WEB SERVICES TO SUPPORT A COMPLEX DECISION SITUATION

In the following sections we explore the architecture and implementation of a specific Web Services application to integrate multiple Web Services that reflect some of the dimensions that we explored earlier. Figure 5 integrates the service chain concept with the architecture for implementation. The left side of the figure focuses on the service chain ideas (that are central to multi-organizational workflows) and the right side of the figure details the architecture of the implementation. Links between the two sides of the figure indicate how the concepts translate to design.

Service chains essentially link a series of value added steps resulting in the required information being delivered at a suitable end point. An organizational event triggers a sequence of information gathering activities along the service chain. Sources of information may exist within or outside the organization; a business process must define the collection of such services that results in the process being executed. We show six equivalences (shown by labeled arrows) between the service chain concept (left-hand side of Figure 5) and the corresponding architecture (right-hand side of Figure 5).

1. An organizational event triggers process execution–this is the initiating point where relevant information needed to commence a sequence of actions is collated.
2. A terminating event indicates the end point of the process –a result is presented to the client.
3. A service may come from within the organization (an internal service)–it is one of the components needed to commence the execution sequence.
4. A service may come from outside the organization (an external service)–it is also one of the required components necessary to commence execution.

FIGURE 5. Service Chain Concept and Architecture



5. An activity transforms information in a value added manner to provide useful information further down the chain.
6. This shows the boundary between the internal and external parts of the focal organization. Note that services may exist on either side of this boundary indicating the multi-organizational nature of the application.

Architecture (See Right Side of Figure 5)

The application accesses, manipulates, and integrates results from multiple Web Services spread across the internet, intranet, and local databases to support a complex decision process (Figure 5). A spreadsheet-driven prototype emphasizes the client side implementation done in a familiar modeling environment. The prototype itself is the client application and the Web Services it consumes act as the server components. The client application can use a number of Web Services to achieve its goal. These Web Services can either be external to the client environment and hosted on the Web (Internet) or can be internal to the client environment (Intranet), or a combination of the two. We refer to the externally accessed Web Services as the "Web server layer." The spreadsheet prototype uses six different methods provided by two different external Web Services (server layer). It also uses two different Web Services hosted internally in the client environment.

The first tier shown in the client layer of the architecture is the presentation tier. This tier presents the prototype as a single integrated entity that provides related functionality. The fact that this functionality is derived from different sources is transparent to the user.

The second tier in the client layer is the proxy tier. A *proxy* is a person (or object) authorized to represent and act for another. The proxy components tier contains localized 'proxy' components. These objects provide a client-side representation of the actual components located at the server. The client application uses these proxies to access both the internal and external Web Services. The proxies themselves are mostly generated by the development environment. The next tier that deals with mapping, interpretation and messaging, along with the generation of the HTTP requests and handling the responses is also dealt with automatically by the development environment. However certain kinds of mapping are still done explicitly by the client application. This sort of mapping deals with linking different Web Services together based on their input and output parameters.

The final tier of the architecture is the data tier, which allows access to locally stored data. The client DBMS is used to store data retrieved from the various external Web Services and also to store data generated locally by the internal Web Services and the application itself. Externally generated data is stored if this historical data needs to be cached locally. If the requested data exists locally then this is used or else it is pulled from the external Web Services. In the case of the spreadsheet prototype, external data in the form of historical stock quote data is stored locally every time the historical quotes Web Service is invoked. This way a check is done to see if the requested data already exists in the database before the Web Service is invoked again.

An execution sequence is triggered by an organizational event that results in the instantiation of relevant objects that relate to that event. In Figure 5, Oonce all of the necessary relevant services have been integrated at step 1 in Figure 5, the user is able to execute the service chain. When the application is executed, at step 2 in Figure 5, a number of actions occur. First, the application finds all of the relevant services in the service chain, and instantiates all the parameters of the data services. Next, the application retrieves the data from the parameters, and sends it to the Web Service. This is accomplished at step 3 in Figure 5.

At this stage a check is done to see if the service is an in-house service or one that is externally accessed on the Web server layer. If the service is an in-house one, then it is locally accessed and data is retrieved from the client DBMS itself (step 4, Figure 5). If the service is being provided by an external Web Service, this Web Service is accessed at the web server layer. Regardless of the service being external or internal, the application updates the client data service with the data returned by the Web Service. At step 6 (Figure 5), the application updates the database that the data service is linked to, with the data returned by the Web Service. The output is also displayed to the user at the presentation layer of the prototype (step 7, Figure 5). A particular instantiation of a business process may be stored as a scenario for later inspection and use. This is a collection of the results from the execution stored as an XML spreadsheet object so that it can be accessed by other external applications through a locally hosted Web Service. Existing scenarios can also be loaded on to the application and further analyzed and modified. This entire process amounts to a series of push and pull operations, where data is pulled from one service and pushed to another. The order of this pushing and pulling defines the business process we are building.

Implementation

This section describes the client-side application delivered through a familiar spreadsheet environment. The application provides for a user to get stock quotes and option values based on the Black-Scholes model. (Black and Scholes 1973) It also provides charts of historical data for the stocks. The data can be viewed on an individual or comparative basis of all the stocks in a spreadsheet environment. All this is achieved through the consumption of Web Services of varying degrees of complexity.

The external Web Services were all sourced from *Xignite* (www.xignite.com). *Xignite* is a company that helps financial institutions deliver wealth management services to investors. They provide consulting services to this end and offer two complementary solutions:

- *xPortal*: A flexible wealth management portal that acts as integration layer deployed on top of the different resources available in the firm.
- *xServices*: A collection of wealth management services in the form of Web Services.

The user interface fundamentally follows the spreadsheet paradigm with some additional functionality provided by buttons on top of the sheet (Figure 6). Columns in the spreadsheet that are populated by data from an external web service are shaded grey. This is the opening screen. The user can either enter the symbols in the "Stock Symbol" column or use a section of "Top Gainers," "Top Losers" or "Top Movers" from either the NYSE, NASDAQ or AMEX stock exchanges. Using this option would give the user a pop-up window from where selections can be made. The user can select the stock symbols from this list which will be added to the existing list in the "Stock Symbol" column (Figure 7).

On clicking the "quote" button, Web Services are called to fill in the stock name and the stock quote. A local service will populate the other columns–Strike Price, Years, Risk-Free Rate and Volatility (Figure 8). These additional columns are populated with default values in calculating the stock option prices based on the stock quotes received. The user can then change these values to calculate the Option price.

The option price is fetched from another Web Service with all the values for a particular stock taken in from the excel sheet. This is done so by clicking the "Options" button and by selecting either a "put" or a

FIGURE 6. Web Services Application Interface

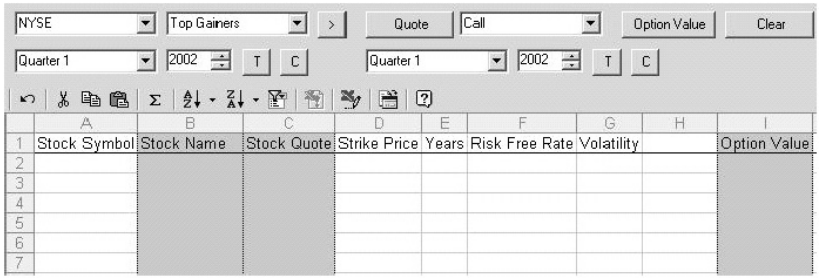
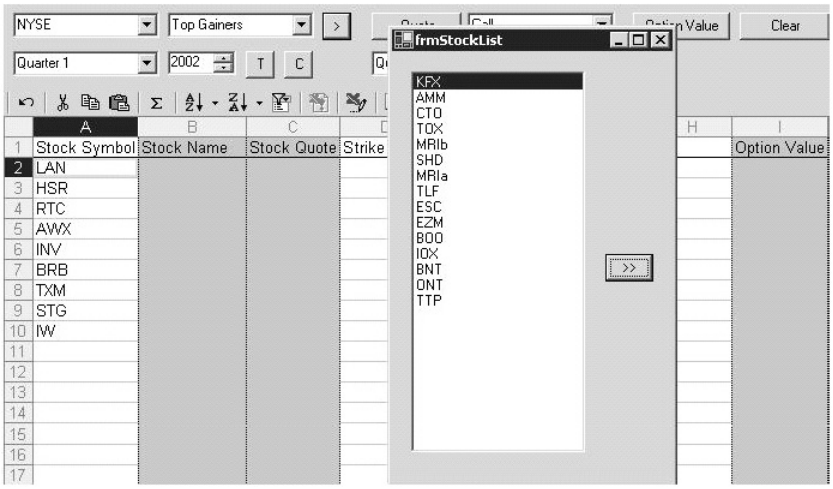


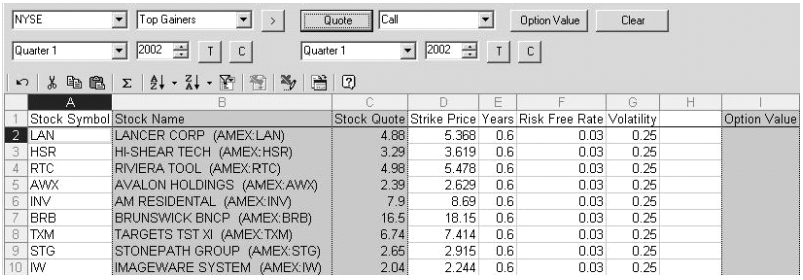
FIGURE 7. Stocks for Consideration



"call" value based on the type of option required to be calculated. This results in the option values being populated (Figure 9).

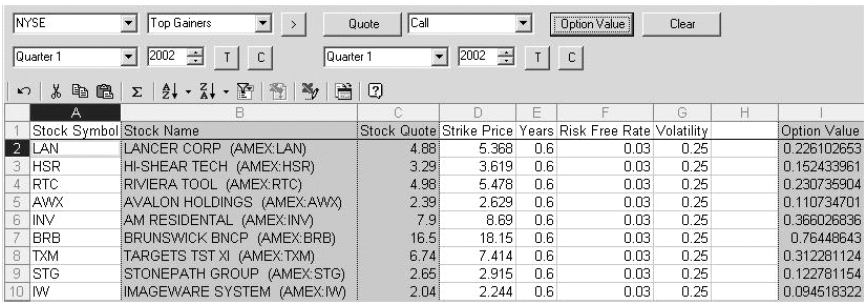
The user can view a chart of historical data based on the type of view that is required. This could either be a chart of a single stock or for all the stocks in the spreadsheet (Figure 10). The Web Services that are required to fetch historical data for the selected period for the individual symbols can be called individually. This could be time consuming if there were a lot of symbols on the list. For this the application makes used of historical

FIGURE 8. Local Web Service Populates Strike Price, Years, Risk Free Rate and Volatility



	Stock Symbol	Stock Name	Stock Quote	Strike Price	Years	Risk Free Rate	Volatility		Option Value
2	LAN	LANCER CORP (AMEX:LAN)	4.88	5.368	0.6	0.03	0.25		
3	HSR	HI-SHEAR TECH (AMEX:HSR)	3.29	3.619	0.6	0.03	0.25		
4	RTC	RIVIERA TOOL (AMEX:RTC)	4.98	5.478	0.6	0.03	0.25		
5	AWX	AVALON HOLDINGS (AMEX:AWX)	2.39	2.629	0.6	0.03	0.25		
6	INV	AM RESIDENTAL (AMEX:INV)	7.9	8.69	0.6	0.03	0.25		
7	BRB	BRUNSWICK BNCP (AMEX:BRB)	16.5	18.15	0.6	0.03	0.25		
8	TXM	TARGETS TST XI (AMEX:TXM)	6.74	7.414	0.6	0.03	0.25		
9	STG	STONEPATH GROUP (AMEX:STG)	2.65	2.915	0.6	0.03	0.25		
10	IW	IMAGEWARE SYSTEM (AMEX:IW)	2.04	2.244	0.6	0.03	0.25		

FIGURE 9. Execution of the Option Price Web Service



	Stock Symbol	Stock Name	Stock Quote	Strike Price	Years	Risk Free Rate	Volatility		Option Value
2	LAN	LANCER CORP (AMEX:LAN)	4.88	5.368	0.6	0.03	0.25		0.226102653
3	HSR	HI-SHEAR TECH (AMEX:HSR)	3.29	3.619	0.6	0.03	0.25		0.152433961
4	RTC	RIVIERA TOOL (AMEX:RTC)	4.98	5.478	0.6	0.03	0.25		0.230735904
5	AWX	AVALON HOLDINGS (AMEX:AWX)	2.39	2.629	0.6	0.03	0.25		0.110734701
6	INV	AM RESIDENTAL (AMEX:INV)	7.9	8.69	0.6	0.03	0.25		0.366026836
7	BRB	BRUNSWICK BNCP (AMEX:BRB)	16.5	18.15	0.6	0.03	0.25		0.76448643
8	TXM	TARGETS TST XI (AMEX:TXM)	6.74	7.414	0.6	0.03	0.25		0.312281124
9	STG	STONEPATH GROUP (AMEX:STG)	2.65	2.915	0.6	0.03	0.25		0.122781154
10	IW	IMAGEWARE SYSTEM (AMEX:IW)	2.04	2.244	0.6	0.03	0.25		0.094518322

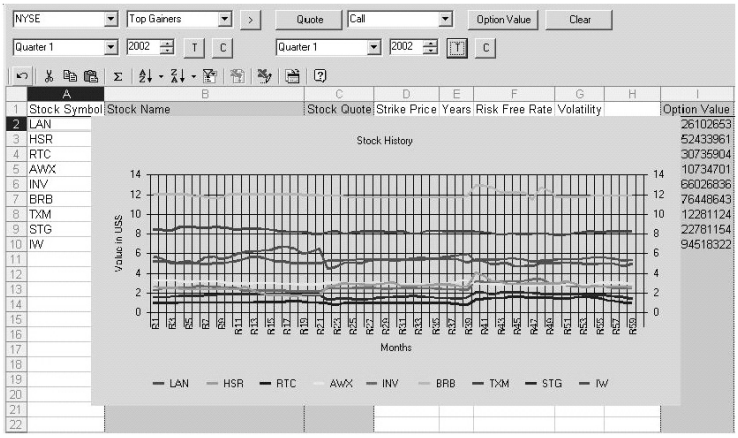
data stored in the local cache. If the requested data exist locally then these are used; otherwise they are pulled from the Web Service.

IMPLICATIONS OF THE FRAMEWORK

The framework we have built supports the variety of Web Services defined in the taxonomy. It is general enough to be applied across many domains. This is because we address the fundamental business problem of integrating heterogeneous and loosely coupled components to form a service chain. These building blocks can be the foundation of a huge variety of systems, much as chains of simple proteins in DNA create a huge and complex array of organisms.

In order for organizations to evolve, they will need to be able to reap the benefits of the new service model. Quickly and dynamically creating

FIGURE 10. Graph of Historical Stock Data



peer-to-peer interactions will allow them to build inter-organizational value chains. Using workflow principles will give users the tools to harness services in this way.

Our framework provides users with these tools, and provides a set of functionality they can use to quickly and easily knit together diverse services. In doing so, support is provided for all the four quadrants of the e-Business service chain grid we developed. The framework supports simple and complex service chains, as well as intra and inter-organizational service chains. It does this in a flexible and modular way. The framework therefore supports not only simple operational business processes, but complex strategic service chains as well. This has implications for enterprise application integration, business process management, value chain management, and e-Business as a whole.

CONCLUSION

Web Services continue to be an important emerging technology. They have important implications for e-Business in particular. We have developed a taxonomy to describe Web Services, as well as a grid to outline how they support e-Business when linked together in service chains. We have also developed a framework and architecture that support the creation and management of service chains. These chains will

form the business processes of organizations as they adopt the service vision. They will also increase the ease of inter-organizational integration in the future. Being able to create and orchestrate service chains using a workflow interface will provide many benefits to users, including increased understanding of processes and higher productivity. Support for all kinds of e-Business, from operational tasks to strategic ones, will be provided by a tool that supports the quadrants of our grid. Our Web Service orchestration prototype is a first attempt at such a tool.

REFERENCES

Abramson, D. J., Komineni, McGregor, J. L., and J. Katzfey. 2004. An Atmospheric Sciences Workflow and its Implementation with Web Services, *The International Conference on Computational Sciences, ICCS04*, Ktakow, Poland, June 6-9.

Aldrich, S. E. 2003. Sun Microsystems, Inc.: Sun Transforms Its Business with the Sun One Platform and Architecture and Java Web Services, Boston, Patricia Seybold Group.

Arkin, A., S. Askary, S. Fordin, W. Jekeli, K. Kawaguchi, D. Orchard, S. Pogliani, K. Riemer, S. Struble, P. Takacsi-Nagy, I. Trickovic, S. Zimek. 2002. Web Service Choreography Interface 1.0, Available: http://www.w3.org/TR/2002/NOTE-wsci-20020808/. Accessed 27 November 2005.

Black, F., and M. Scholes, 1973. The Pricing of options and corporate liabilities, *Journal of Political Economy*, 81 (May/June): 637-654.

Curbera, F., Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana. 2002. Business Process Execution Language for Web Services 1.0, Available: ftp://www6.software.ibm.com/software/developer/library/ws-bpel1.pdf. Accessed 27 November 2005.

Davenport, T. H., and J. E. Short. 1990. The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, 31(4), 11-27.

DISA. 2005. Defence Information Systems Agency, http://www.disa.mil/main/about/missman.html. Accessed 27 November 2005.

Dustdar, S. 2004. Web Services Workflows–Composition, Co-ordination, and Transaction in Service-orientated Computing, *Concurrent Engineering Research and Applications*, 12(3), 237-246.

Ferris, C., and J. Farrell. 2003. What are Web Services? *Communications of the ACM*, 46(6), 31.

Fremantle, P., S. Weerawarana, and R. Khalaf. 2002. 'Enterprise Service', *Communications of the ACM*, 45(10), 77-82.

Gottschalk, K. 2000. Web Services Architecture Overview: The Next Stage of Evolution for e-Business, IBM developerWorks, http://www-106.ibm.com/developerworks/library/w-ovr/. Accessed 27 November 2005.

Gottschalk, K., S. Graham, H. Kreger, and J. Snell. 2002. Introduction to Web Services Architecture, *IBM Systems Journal*, 41(2), 170-177.

Gralla, P. 2003. Deployment Profile: Yellow Transportation Delivers Three Web Services, The Web Services Advisor.

IBM. 2003. Web Services Case Studies: Defense Information Systems Agency, IBM, http://www-306.ibm.com/software/ebusiness/jstart/casestudies/disa.shtml. Accessed 27 November 2005.

Johnston, S. J. 2002. State of Web Services, InfoWorld, http://www.infoworld.com/articles/pl/xml/02/02/04/020204plwebstate.xml. Accessed 27 November 2005.

Lawrence, P., Ed. 1997. Workflow Handbook 1997, Workflow Management Coalition, New York, John Wiley and Sons.

Leymann, F., D. Roller and M. Schmidt. 2002. Web Services and Business Process Management, *IBM Systems Journal*, 41(2), 198-211.

McKeen, J. D., and H. A. Smith. 2002. New Developments in Practice: Managing the Technology Portfolio, *Communications of the AIS*, 9(5), 1-25.

Nghiem, A. 2002. IT Webservices: A Roadmap for the Enterprise, Prentice Hall, USA.

Piccinelli, G. and S. L. Williams. 2003. Workflow: A Language for Composing Web Services, Lecture Notes in Computer Science, H. A. Reijers. Berlin, Springer-Verlag.

Reijers, H. A., Ed. 2003. Design and Control of Workflow Processes, Lecture Notes in Computer Science. Berlin, Springer-Verlag.

Stal, M. 2002. Web Services: Beyond Component-based Computing, *Communications of the ACM*, 45(10), 71-76.

Tanriverdi, H. 2005. Information Technology Relatedness, Knowledge Management Capability, and Performance of Multibusiness Firms, *MIS Quarterly*, 29(2), 311-334.

van der Aalst, W. M. P. and A. Kumar. 2001. XML Based Schema Definition for Support of Inter-organizational Workflow, Bell Labs.

RECEIVED: July 1, 2005
REVISED: March 30, 2006
ACCEPTED: March 31, 2006

doi:10.1300/J179v05n04_02