The potential of Webservices to enable Smart Business Networks

Jos van Hillegersberg
jhillegersberg@fbk.eur.nl
Rotterdam School of Management
Erasmus Universiteit Rotterdam, the Netherlands

Ruurd Boeke
r.boeke@sitechno.com
Sitechno, The Netherlands

Willem-Jan van den Heuvel
W.J.A.M.vdHeuvel@uvt.nl
InfoLab, Tilburg University
The Netherlands

1 Introduction

After the hype, the web still holds huge potential for B2B integration. After a majority of companies have realized some sort of, mostly static, web presence, the web’s new promise is to enable smooth and cross-organizational business integration. Webservices seem to be among the key web technologies that will allow this to happen. Eventually, these technologies should enable the transformation of current static supply chains into dynamic virtual networks of enterprises.

Proponents of Webservices technology frequently claim that Webservices will lower barriers for “plug and play” B2B integration. Through webservice oriented architectures, current supply chains could become better integrated, more agile and eventually intelligent. Others have said that these technologies are not yet ready for large scale applications to supply chains and propose “traditional” cross-enterprise integration methods such as centralized E-Hubs which are able to bridge technology gaps through custom made adapters and (semi-) automatic translations.

Unfortunately, extensive reports from practice to settle this debate are lacking. Dynamic business networks enabled through Webservices technologies are not yet a reality. The vast majority of current practical work is applying webservices to improve intra-organizational integration rather than connecting different businesses into a network. Most research and industry reports in this area focus on new technologies and standards, usually adding to and refining the webservice technologies. Currently, new technologies, tools and standards related to webservices are being introduced at a high pace. ICT companies, consultancies and industry consortia are the main proponents of webservices solutions. Few research addresses the potential benefits to current supply chains or addresses implementation issues. There is a lack of experience reports, case studies, demos, simulations and sample applications concerning the application of recent webservice technologies to illustrate and evaluate how Webservices could transform static and inefficient supply chains into more efficient and dynamic “smart” business networks.

This paper attempts to start filling this gap by focusing on the application of Webservices technologies to enable dynamic and smart business networks. The paper will address the following research questions sequentially:

*Question 1:* What are the current and near-future standardized components of webservice technology relevant to enabling smart business networks? (section 2)

This question we address through a literature review and by evaluating tools, technologies and methods for webservice development.
Question 2: What is the current state-of-the-art in web services to enable smart business networks? (section 3)

This question we address through developing a typical scenario in which we transform a static supply chain scenario to a ‘loosely coupled’ business network. We implement the scenario using state-of-the art “enterprise application integration (EAI)” and Web services orchestration technology.

Question 3: What are research issues regarding the use of Web services technology to enable smart business networks? (section 4)

This question we deal with by reflecting on our experiences with the development of the scenario. From our development experiences we deduct research issues and agenda for Web services technology in the light of its application to enable smart and dynamic business networks.

2 Web services

The Service Oriented Computing (SOC) paradigm (Papazoglou, 2003), with as its main materialization web-service technology, promises to deliver a profound new way of developing business applications and a significant step forward in the quest to maximize software reuse and integrate systems across different technology platforms. Although a variety of definitions of SOC exist, key to SOC is the process of constructing software application by way of orchestration of pre-fabricated and pre-tested Web-Services (WSs).

WSs can be defined as self-describing, interoperable and reusable business components that can be published and invoked through the Internet, even when they reside behind a company’s firewall. WSs constitute in fact both design- and runtime, platform-agnostic distributed enterprise building blocks that can be composed into higher-order assemblies that support (inter- or intra-) organizational business transactions. Hence, web-service enable enterprises to leverage massively distributed applications, cutting across “traditional” supply chains and vertical industries.

WSs are considered as an important candidate to overcome some severe obstacles of CBD. In particular, WSs are described in a standard manner by their interface separating value-adding commodities from the actual implementation. WS employ open, text-based standards to achieve interoperation between applications that possibly reside at various collaborating organizations. While doing so, SOC overcomes one of the major disadvantages of object middleware technologies such as CORBA, which in fact were too complex, firewall unfriendly and hard to maintain. In addition, SOC puts greater emphasis on three complementary views on services: a client view, a provider view and a broker perspective, witnessing the emergence of a Service Oriented Architecture (Burbeck, 2000). Software components usually needed to be licensed and integrated on the customer’s site. Webservices reside at the suppliers site, thus supporting a federated architecture. The customer buys access to the WS instead of a license to download and integrate software code or a component. This model delivers potential benefits similar to application service provision (ASP). These include automatic upgrades to new versions, clear contracts through service level agreements and often specialized support from the vendor, etc. Lastly, services extend traditional class interfaces by including information that is vital to setup Service-Level Agreements by attaching information about extra-functional properties (UDDI, 2003), like performance, security, and pricing information.

The first wave of web-service technology standards has been devoted to developing infrastructural solutions for achieving interoperation at the level of messaging middleware. By now, industry has reached common agreement about interfacing standards for WSs using
W3C’s Web Service Description Language (WSDL, (Christensen, 2001)) that builds top of W3C’s Simply Object Access Protocol (SOAP). These technologies and standards allow organizations to expose their services and lookup and invoke services of other organizations using the Internet. XML-based standards are available or being developed to invoke services (SOAP), define services (WSDL) plus a huge number of sector specific standards, store services in a registry (e.g. ebXML registry, UDDI). Although these initial WS standards enable simple transactions using basic describe, publish, interact mechanisms, these are not sufficient to support the more complex and long transactions that take place in a business network.

Therefore, several industry consortia are now developing higher-order web-service stacks on top of SOAP that provide process composition languages (e.g., BPEL4WS (Cabera,2002), transactional ACIDity (BTP, 2002), WS-Transaction, (Cabera,2002a), and Security (WS-Security)). Figure 1 illustrates the WS standards stack and some of these standards.

![Figure 1: webservices standards stack](Source Curbera et al., 2003)

In this paper we focus on the possibilities of the prominent current WS standards transport (SOAP,XML), description (WSDL) and the recently proposed business process standard BPEL4WS and explore their potential to support dynamic business networks.

Several comparable standard proposals have preceded BPEL4WS (e.g. WSFL by IBM, XLANG by Microsoft and WSCF by HP). Also, there are currently several competing standards (e.g. BPML by BPML.org) , or proposals with a different focus (e.g. WSCI by Sun, SAP, BEA, and Intalio). These standards have comparable features, and as Peltz (2003) notes: “…it is still very unclear which ones will emerge as industry standards for web services orchestration and choreography. There are key advantages to each standard”. However, Peltz further notes that BPEL is backed up by major players in the industry (Microsoft, IBM, Siebel Systems, BEA, and SAP coauthored version 1.1 of the BPEL4WS specification) and is already widely supported by development tools. Therefore, we have looked at BPEL for our explorative study and provide a brief overview of the standard now (see Cabera, 2002 for details).

Cabera (2002) summarizes BPEL as follows: “BPEL4WS defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces, and the structure of the relationship at the interface level is encapsulated in what we call a partner link. The BPEL4WS process defines how multiple service interactions with these partners are coordinated to achieve a business goal, as well as the state and the logic necessary for this coordination. BPEL4WS also introduces systematic mechanisms for dealing with business exceptions and processing faults. Finally, BPEL4WS introduces a mechanism to define how individual or composite activities within a process are to be compensated in cases where exceptions occur or a partner requests reversal” .
As observed by Peltz (2003), BPEL is built upon, and heavily uses WSDL. Every BPEL process is exposed itself as a webservice described in WSDL. The BPEL process uses WSDL datatypes and WSDL to call external webservies required (see Figure 2).

BPEL supports both WS-orchestration and WS-choreography. Peltz (2003) describes the difference between the two as; “Orchestration refers to an executable business process that may interact with both internal and external web services. For orchestration, the process is always controlled from the perspective of one of the business parties. Choreography is more collaborative in nature, in which each party involved in the process describes the part they play in the interaction.”

For our exploratory study, we focus on WS-orchestration. A WS-orchestration allows a party in a business network so quickly define and execute a business process incorporating the services of business partners in the network. Therefore, WS-orchestration technology seems especially useful for enabling dynamic business networks.

The actual shape and possibilities of BPEL combined with the early webservice standards SOAP and WSDL will receive more attention in the next section when we implement a business network scenario.

3 Implementing a business network scenario enabled by Webservices

To explore the potential of the current state-of-the-art in webservies to enable smart business networks we now develop a typical scenario in which we transform a static supply chain scenario into a ‘loosely coupled’ business network that can be easily adapted and configured. We implement the scenario using state-of-the-art “enterprise application integration (EAI)” and Webservices orchestration technology.

The scenario used here is partly based on a supply chain scenario developed by the Webservices Interoperability organisation (WS-I.org). WS-I is an open, industry organization chartered to promote Web services interoperability across platforms. WS-I has developed UML specifications of a Supply Chain Management scenario that for the purpose of evaluating WS standards and tools. The specification documents includes Use Cases, Use Case Scenarios, Activity Diagrams and a Sample architecture. Several vendors have built sample WS-applications using these specifications including Microsoft, IBM, Novell, BEA and SAP (WS-I, 2004).
The WS-I sample supply chain scenario is based on a typical B2C model in which Retailer receives customer orders and has to fulfil these orders by ordering from a collection of warehouses. The retailer has to manage stock levels in warehouses and replenish by ordering from a Manufacturer’s inventory (a typical B2B model). In order to fulfil a Retailer’s request a Manufacturer may have to execute a production run to build the finished goods (WS-I, 2004). Figure 3 shows part of the WS-I scenario (Sourcing of goods from a Warehouse) in an UML Activity Diagram. The process flow shows that ordered goods are located and shipped from warehouse(s). The objective of this sourcing process is to deliver all goods ordered and prevent “out of stock”

Figure 3: Sourcing process of a Retailer in the WS-I sample supply chain scenario (WS-I, 2004)

The WS-I scenario is a useful basis for our study. However, it is quite broad and models a static supply chain. For example, the number of warehouses is fixed (3). Moreover, the business rules are very basic. For example, as figure 3 illustrates, orders are shipped from the warehouses only based on a fixed sequence checking the available stock at warehouse A, B and finally C, thus not comparing cost information.

Therefore we have developed our own scenario, which is based on the WS-I supply chain scenario, but adds more dynamics to the business network and slightly more complexity to the business rules. In the scenario, consumers also can order products from a ‘simulated’ webshop (see figure 4). The consumer can order multiple products in a single session.
Figure 4: sample webshop application

The shop checks the warehouses through calling their webservices to inquire about stock levels. Similarly to the WS-I scenario, the sourcing process will sequentially check multiple warehouses if the stock of a single warehouse is not sufficient. Differing from the WS-I scenario is the option to change the number of warehouses and the product assortment of each warehouse. The scenario can thus be used to experiment with “smart” webservices orchestrations that use more complicated business rules to simulate the efficiency of order delivery for various business networks and warehouse product assortments (Figure 5).

Figure 5: Managing the Warehouses and Product Assortment in the business network

The Warehouse system stores product ID’s, prices and stock levels and replenishes stock by ordering from a producer whenever the stock level drops below a minimum level. The Warehouse system keeps track of the number of items in stock, the number of items ordered by customers and the number of items in backorder with a producer (status awaited). Other than in the standard WS-I scenario, prices can be different for the same product in different warehouses (Figure 6). Using this information, smart Webservice orchestrations can be devised that optimize in time delivery and aim for the lowest price.
Figure 6: Managing the Warehouses and Product Assortment in the business network

Finally, producers are included in the scenario. Producers can be configured to produce certain products in certain batches. The warehouse can order from any producer in the business network. At this stage, the emphasis is on enhancing the Retailer-warehouse relationship.

Initially, the scenario was implemented in the Microsoft.Net framework. Using standard Enterprise Application Integration techniques (like Remote Method invocation), communication between webshop, warehouse and producer was established. Then, the scenario was ported to Webservice technology. By using a WS orchestration tool (BizTalk 2004) and BPEL webservice orchestration, the B2B processes were replaced by WS-orchestration. These were previously 'hard coded' into the .net (C#) code. This exercise allowed us to clearly compare the potential of WS-orchestration languages and tools as compared to traditional Enterprise Application Integration (EAI) techniques. In the following discussion, we focus on the sourcing process between the Webshop and the retailer.

In porting the standard EAI scenario to a WS-orchestration, we had the following questions in mind. First, what is the power of BPEL to support more complex B2B interactions? Would it be limited to modelling simple straightforward interactions or could it also be used to quickly setup more complex and long business transactions? Second, how mature and compatible are the standards and tools. As mentioned in the previous section, orchestration standards such as BPEL as still under development. Does this cause unexpected side-effects and unpredictable behaviour of the supporting tools? Finally, and most importantly, will WS-orchestration technology enable the end-user to quickly setup and adapt B2B processes? Will WS-orchestration technologies finally allow business users to escape from costly, and error prone traditional EAI technologies that they did not understand? In other words, will these technologies soon empower the end-user to become a “business network supervisor and configurator”?
The implementation project started in 2003. Initially, we used Microsoft BizTalk 2002 as the WS-orchestration product. However, the product had several shortcomings, and when BizTalk 2004 was launched in February 2004, we decided to migrate to this product. We have considered several other tools that supported WS-orchestration for this study, such as tools from Collaxa, BEA, IBM and Cordys. However, comparisons of several of these products (see Peltz, 2003) illustrate that differences are minor and for our purposes all of these tools could have been used.

BizTalk 2004 includes a several components such as a visual orchestration designer, a ‘message-mapper’ to graphically translate and map different XML messages and a Business Rule composer that is meant to isolate business rules from the application. The BizTalk monitoring environment contains a performance measurement tool and a debugging tool. To get a feel for the usability and adaptability of BPEL WS_ orchestrations in BizTalk, we will briefly discuss the WS-orchestration between the webshop and the Warehouses.

![Image](image.png)

Figure 7: Start of the Webshop-Warehouse orchestration in BizTalk 2004

As mentioned earlier, a BPEL WS orchestration is itself a Webservice. Therefore, the orchestration starts with a incoming request, which is an customer order requesting some quantity of a product. The orchestration registers monitors a ‘port’ for incoming requests and starts an orchestration whenever an order message is received.

As our scenario uses a dynamic number of warehouses which have unique ID’s, we built separate webservices to locate Warehouses ID’s (a kind of simple directory service). In a truly dynamic business network, indeed, an orchestration process could decide to check directory services to short-list potential partners in a business network. Once the list of Warehouses to check for stock has been identified, the orchestration can start checking product availabilities at each warehouse.
Figure 8: A loop including external service requests to select a Warehouse for Sourcing

Figure 8 shows the key part of the BPEL WS-orchestration. The connection to the Warehouse Webservice is quickly established. An external ‘port’ can be created by simply specifying the Web-address of the Webservice and the request and response messages are connected graphically by using ‘drag and drop’ to the node ‘avail request’. The orchestration shown will loop through all selected Warehouses and check for warehouses that have enough stock to fulfil the order in a single batch (through an IF-THEN construct which is visually shown by the diamond shape in Figure 8). The loop will store the details of this warehouse and the product price.

After all Warehouses have responded the loop ends and the remainder of the orchestration can implement the order strategy and handle the case that not enough stock is available at a single Warehouse. We have implemented several variants to test the adaptability of the orchestration language and tools.
4 Conclusion and Discussion: an R&D agenda for webservices to enable smart business networks

Webservices are a promising technology to enable dynamic and smart business networks. However, the current WS stack of standards can no longer be considered a set of simple and coherent technologies as in the early days of WS. The technology is maturing, and as issues such as WS orchestration, choreography, security, quality of service and transactions are resolved, the set of standards gets richer and naturally also more complex. An area in which standards are still evolving and competing is the area of WS orchestration. In the last couple of years, we have seen many different standards rising and disappearing, and given this high technological pace, it is not surprising that businesses are reluctant to largely invest in this part of WS technologies, and examples of business networks that already use orchestration technologies are scarce.

Therefore, to evaluate the state of the art and its potential to enable smart and dynamic business networks in the future, implementing scenario’s is a useful research method. Work as done by the WS-I to develop standard sample specifications to serve as a test-bed, interoperability test and reality check for the new WS-technologies is extremely useful. A downside of such initiatives is that vendors are providing sample solutions, and objective reviews of the possibilities of the tools and standards are scarce.

In this study we aimed to evaluate the current state-of-the-art in webservices to enable smart business networks. From the scenario we developed and test implementation using WS-orchestration several conclusions can be drawn. First, the well established and relatively simple SOAP and WSDL standards enable a true cross-platform distributed architecture. As long as these standards are well supported across platforms, Webservices will truly allow straightforward B2B integration using standard and low-cost internet technology. This is a major advantage in enabling business networks, as small companies within these networks usually do not have the knowledge, time and money to implement traditional and complex EAI technologies.

Second, our scenario implementation clearly demonstrated that the network orchestration could be designed mostly separately from the various systems available in the business network. Thus, the network business rules (such as selecting a warehouse to order from) are isolated from the ‘back-office systems’ and therefore more ‘visible’ and easier to change. This is a key advantage to creating dynamic business networks. Although it is hard to define what is exactly meant by a “smart” business network, the example also illustrated that basic decision logic could be easily built into an orchestration.

Third, we found that the orchestration technology has greatly advanced over the last 2 years that we were carrying out this project. Still, although it has become much easier to put together orchestrations, it is in our opinion still the work of specialized consultants/developers. It is more likely that after some training, non-IT staff could adapt the business rules within an orchestration. However, businesses should handle these changes with care, as the impact of malicious code in a WS-orchestration will not be limited to the internal business, but will also impact suppliers, partners and even worst, customers.

Our scenario has not tested performance, security and scalability of the orchestration tools, nor were the B2B processes complex enough to reach the limits of the expressive power of BPEL.

Moreover, as business networks often lack a “chain director” we claim that languages and tools to design B2B interaction should have features to support collaboration. Intelligence is required as the coordination cost (searching for appropriate services, negotiating for price and quality of services, monitoring B2B transactions) may become too high in dynamic business networks. Therefore, intelligence should be added to web-service enabled business networks. This is where research on agent technologies in supply chains and webservices technology are likely to meet in the future. It is our first assumption that both technologies are complementary and developing agents is more feasible in a B2B network that utilizes webservices standards.
Nevertheless, the progress in WS-orchestration technologies has been substantial recently, and the scenario implemented in this paper has demonstrated that enabling a business network through WS-orchestration is very well feasible. Real benefit is only expected in cases where supply chains can benefit from transforming to a dynamic network of businesses. Especially when quick adaptability of the business network is an important issue, the use of WS-and WS-orchestration should definitely be considered.

5 References

(UDDI, 2003), http://www.uddi.org/
(Ws-I.org, 2004), http://www.ws-i.org/