African Journal of Business Management Vol. 4(13), pp. 2604-2614, 4 October, 2010
Available online at http://www.academicjournals.org/AJBM
ISSN 1993-8233 ©2010 Academic Journals

Review

Enterprise Application Integration (EAI), Service Oriented Architectures (SOA) and their relevance to e-supply chain formation

Mohammad Rehan[1]* and Goknur Arzu Akyuz[2]

[1]Department of Computer Engineering, Atilim University, Kızılcasar Mahallesi, 06836 Incek Gölbasi, Ankara, Turkey.
[2]Department of Industrial Engineering, Atilim University, Kızılcasar Mahallesi, 06836 Incek Gölbasi, Ankara, Turkey.

Accepted 28 July, 2010

The greatest challenge of e-supply chain formation is the integration of cross-application processes in a seamless manner. As platform independent, web-based interoperability are becoming key issues in web-based integration and supply chain formation, organisations are in search of architectural solutions to overcome these challenges. Thus, this paper focuses on the service oriented architectures (SOA) as the recent trend in cross-platform enterprise application integration. SOA paradigm is discussed in detail with special focus on e-supply chain formation. Findings indicate that SOA still appears as the most convenient paradigm to meet the challenges of today's e-supply chain formation requirements.

Key words: EAI (Enterprise Application Integration), SOA (Service Oriented Architecture), e-supply chain.

INTRODUCTION

Modern businesses need functionality that is both distributed and centralized. Existing systems, such as ERP (Enterprise Resources Planning), CRM (Customer Relationship Management) and SCM (Supply Chain Management) serve the needs of key segments of the organisation. Woods and Mattern (2006) emphasise the need for a flow moving from one "system of record" to another, with the context for the process kept outside of any of the existing systems. The traditional way of building enterprise software is not well-suited to these new requirements and does not take full advantage of the new world of networks, reusable services, and distributed data. Utilizing already present applications in a"self-contained, isolated manner "is not sufficient and the real challenge facing companies is the integration. The real question put forward is: "How could all of the best-of-breed applications be made to work together to serve the needs of the cross-application processes that were becoming the key to increased efficiency and innovation?" (Woods and Mattern, 2006). The proliferation of enterprise applications made integration of applications as important as the functionality of the applications themselves (Woods and Word, 2004). As such, the real problem of information technology (IT) departments is to obtain an end-to-end business view of complex business processes by the integration of critical business systems such as CRM and ERP, which often operate in an isolated manner although they span multiple applications (Microsoft Whitepaper, 2006).

This paper focuses on the importance of SOA in enterprise application integration and e-supply chain formation. The methodology selected in this study is the review of recent literature on SOA and available integration platforms. After, defining enterprise application integration and discussing various classifications for EAI in literature, Section 2 mentions the evolutionary shift towards SOA in application integration and the basic technologies. Section 3 discusses the basic concepts of SOA paradigm. Section 4 emphasizes commonalities and basic design principles of available application integration frameworks. Section 5 focuses on the benefits of service

*Corresponding author. E-mail: mdrehan11@gmail.com. Tel: +90 (312) 586 8340.

oriented systems. Section 6 discusses their relevance for e-supply chain formation, section 7 provides discussion and Section 8 concludes.

ENTERPRISE APPLICATION INTEGRATION: RISE OF SOA

Enterprise application integration is defined as "the process of tying together multiple applications to support the flow of information across multiple business units and IT systems" (Sweat, 1999). EAI is about interoperability and information synchronization across multiple applications (mainframe, packaged or purchased systems, and custom application systems), enabling sharing of information, not just within an enterprise or organisation but within a business environment that includes a company, its suppliers and its customers. Enterprise integration occurs when there is a need in improving interactions among people, systems, departments, services, and companies (in terms of material, information or control flows (Vernadat, 2007).

The need for developing systems in heterogeneous environments to accommodate an endless variety of hardware, operating systems, middleware, languages and data stores are clearly mentioned (Bih, 2006); messaging, connectivity and security being as the basic services needed of an EAI solution (Pan, 1999). The need for interoperability is clear for seamless connectivity (Vernadat, 2007) and Legner and Vogel (2008) emphasize that interoperability requires agreements and standardizations at pragmatic, semantic, syntactic, communication and transport levels.

A holistic approach to integration is therefore necessary that takes into account the business strategy as defined from the enterprise vision, the business process definition and enactment, and the design and operation of interoperable enterprise systems as supported by a relevant and efficient IT infrastructure (Vernadat, 2007). As such, handling the business processes and business logic are just as important as the data management side of the issue. Handling the people integration and user interface issues appears as another critical issue to be performed independently of the data and process integration to allow for customized access to shared business data and logic. Microsoft emphasises that "an integrated supply chain requires connecting systems from the factory floor to the storefront and across trading partners, facilitating business processes that span systems, and people, within and across organisational boundaries". This further streamlines the critical business-to-business processes by automating decisions and providing real-time visibility, covering legacy systems integration from different parties as well.

Within the context of EAI, literature reveals various integration levels. Table 1 provides the basic characteristics of these classifications.

EAI has been a top issue since 2002, along with the e-business transformation, CRM, supply chain management and logistics. The issue is complex since it requires both internal and external integration of many incompatible systems (Bolloju and Turban, 2007).

Regarding EAI, many different technologies emerged so that a cross-application, integrated view of enterprise applications was created, based on the new possibilities of the Internet and emerging technology standards such as HTTP (HyperText Transfer Protocole) and XML (Extensible Markup Language). These new technologies started to bridge the gap among isolated enterprise applications and enabled some cross-application coordination and development, however bringing in a new set of problems: "integration of the integration technologies". Many of the solutions are reported to have limitations or problems including: Being proprietory, platform-dependant or vendor-dependant, requiring significant financial and technological resources, being inflexible (Bolloju and Turban, 2007), assuming synchronous operations and not scaling up (Vernadat, 2007). As such, current solutions are still struggling with overcoming these problems.

Woods and Mattern (2006) mention portals, data warehouses, EAI technology, business process management applications and application servers as the technologies for integration. LaFata and Hoffman (2004) also supports these components, mentioning that Enterprise Application Integration software suites evolved to handle the complex requirements of application integration, providing the following key areas of functionality:

1. An integration broker providing a set of services for message transformation and intelligent routing.
2. Development tools for specifying transformation and routing rules and for building adapters into applications.
3. Off-the-shelf adapters for popular enterprise packaged applications (e.g. SAP R/3).
4. Monitoring, administration and security facilities.
5. Message Oriented Middleware (MOM).
6. Business process managers, e-commerce features and portal services.

Güner (2005) mentions the following technologies in her comparative study: Java Message Service (JMS), Remote Method Invocation (RMI), Component Object Model (COM), Distributed Object Model (DCOM), Common Object Request Broker Architecture (CORBA), Web Services. Basing on Güner (2005), Akyuz (2008) argues the following:

1. In terms of model development, only Web services provide component oriented service development, whereas all others support object oriented approaches.
2. In terms of interface definition language, JMS and RMI have Java dependency and COM/DOM has Microsoft Interface Definition language dependency. As using WSDL, web services represent the language independency.
3. Platform independence is present only for web services

Table 1. Integration levels by various sources.

Source	Classification
Vernadat (2007)	Physical/ application/business
Ibm.com (2007)	Process/data integration
Woods and Word (2004) SAP NetWeaver	People/ information/process/application levels
Bih (2006)	interaction/application connectivity/process integration/ information integration levels
Himalayan (2004)	Data/application /process levels
Ciol.com (2002)	Data/method /user interface levels

Table 2. Evolutionary steps towards SOA.

Evolutionary step	Description
Monolithic	Large scale applications using a procedural coding methodology.
Structured or object oriented	Dividing applications into units of logic based on functionality. The first step of SOA.
Clients and servers	The logical progression of object orientation- bundling groups of functions on the server and invoking them from client.
3-tier	Adding an extra layer to interaction.
N- tier	Layered request-response calls between applications. Portal development relied on this concept.
Distributed objects	Heterogeneous system of many distributed objects.
Components	Aggregating objects into logical components that achieve specific functionality and creating interfaces to these components.
Service oriented Components	An environment of components interacting in a peer-based environment using interfaces based on widely accepted standards to offer services.

Source: Based on Nickull (2005).

services.
4. Interoperability and support for open standards are present only for web services.
5. Only web services support both synchronous and asynchronous modes of communication.

As such, among these technologies, web services represent highest degree of interoperability, platform independency and standardisation. Loose coupling, UDDI and WSDL support characteristics of web services also put the technology ahead of the others. SOA is a new wave for building agile and interoperable enterprise systems (Vernadat, 2007) and one of the most important technological trends in contemporary business organizations (Lior and Seev, 2009).

A brief survey on the history of e-commerce and integration frameworks also reveals a clear shift from client/server technology towards integrated and adaptable businesses basing on web services with changing business requirements and technological advances. Güner (2005) mentions this trend by saying “nowadays, approach to application integration is moving from information oriented to service oriented systems”. This evolutionary transition, starting with monolithic structures towards service oriented components, is also supported by (Nickull, 2005), mentioning the following evolutionary steps for the enterprise systems, given in Table 2.

The shift towards SOA is also evident in various integration platforms developed by proven vendors, like IBM, Microsoft, Oracle and SAP, who provide clear commitment to SOA transition in their products (Gartner Research, 2007). Microsoft BizTalk, IBM WebSphere, SAP NetWeaver and Oracle Fusion MiddleWare integration platforms are among the major ones in this regard, as also detailed in the study on available application integration frameworks above. Hündling and Weske (2003) and Bolloju and Turban (2007) emphasizes this massive SOA support of major software vendors and stress the fact that web services technology is the first major technological approach that vendors like IBM, Microsoft and BEA systems join forces on common standards. The study mentioned is also a clear evidence of the rise of SOA in today's major propriatory solutions.

SOA PARADIGM

SOA is an evolution of the component based architecture, interface based design, distributed systems and the Internet in general (Nickull, 2005). SOA refers to the architecture for software systems in which services are the fundamental building blocks, to any system that exposes its functionality as services (Earl, 2005). SOA is an emerging approach addressing the requirements of loosely coupled, standards-based, and protocol-independent distributed computing (Papazoglou and Heuvel, 2007). Typically, business operations running in an SOA comprise a number of invocations of these different components, generally in an event-driven or asynchronous fashion reflecting the underlying business process needs. To build an SOA, a highly distributable communications and integration backbone is required. This functionality is provided by the Enterprise Service Bus (ESB) that is an integration platform utilizing Web services standards to support a wide variety of communications patterns over multiple transport protocols and delivering value-added capabilities for SOA applications (Papazoglou and Heuvel, 2007).

The basic idea of SOA paradigm is that a system is designed and implemented using loosely coupled software services with defined interfaces that can be accessed without any knowledge of their implementation platform. By overcoming interoperability limitations, SOA allows existing software systems to be integrated by exploiting the pervasive infrastructure of World Wide Web (Canfora et al., 2007). As such, service orientation is a means for integrating across diverse systems. Each IT resource, weather an application, system, or trading partner, can be accessed as a service. These capabilities are available through interfaces (Candido et al., 2009). However, complexity arises when service providers differ in their operating system or communication protocols, causing inoperability. Service orientation uses standard protocols and conventional interfaces (usually Web services) to facilitate access to business logic and information among diverse services. SOA allows the underlying service capabilities and interfaces to be composed into processes. Each process is itself a service, one that offers up a new, aggregated capability. Because each new process is exposed through a standardized interface, the underlying implementation of the individual service providers is free to change without impacting how the service is consumed (Microsoft, 2006). As such, service orientation is an approach to organizing distributed IT resources into an integrated solution that breaks down information silos and maximizes business agility. The approach modularizes IT resources, creating loosely coupled business processes that integrate information across business systems. Critical to a well-designed service-oriented architecture is producing business process solutions that are relatively free from the constraints of the underlying IT infrastructure (Vernadat, 2007) to enable the greater agility that businesses are seeking. As such, Service Oriented Architecture (SOA) ultimately enables the generation of new dynamic applications, sometimes called composite applications (Microsoft, 2006).

To enable composite application generation free from underlying IT infrastructure, SOA utilizes the concepts of Services, Web services, Enterprise services, Process orchestration, Process Choreography (Güner, 2005; Vernadat, 2007; Kunti et al., 2007; Candido et al., 2009; Cheng et al., 2010). The basic characteristics of the "services "concept are being discoverable and dynamically bound, self-contained and modular, location-transparent, interoperable, loosely coupled, having a network-addressable interface, having coarse-grained interfaces (Güner, 2005).

In SOA, web services describe a specialized type of software designed to support a standardised way for provision and consumption of services over the Web, through the compliance with open standards such as XML, SOAP (Simple Object Access Protocole), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration) (OASIS, 2005). Web services are designed to support interoperable machine-to-machine interaction over a network (Güner, 2005). Therefore, Web services are a standard way of creating a self-describing service based on XML that uses the Internet to communicate, where a service is a program that talks to other programs.

Web Services, unlike traditional client/server systems, are not meant for direct end-user consumption. Rather, Web Services are pieces of business logic, having programmatic interfaces and through these interfaces; developers can create new application systems (OASIS, 2005). The motivation behind Web Services is to facilitate businesses to interact and integrate with other businesses and clients, without having to go through integration design and/or to expose its confidential internal application details. This is made possible by "relying on the non-platform dependent and non-programming language dependent XML" to describe the data to be exchanged between businesses or between the business and its clients, using a WSDL to specify what the service is providing; using a UDDI to publish and locate who is providing the service; and typically using SOAP over HTTP to transfer the message across the internet (OASIS, 2005; Candido et al., 2009).

Akyuz (2008) discusses the clear distinction between

Web services and enterprise services, emphasizing that enterprise services are new service definitions developed, typically a series of Web services combined with business logic that can be accessed and used repeatedly to support a particular business process. In this context, a web service is just a standardized interface to a service's functionality whereas an enterprise service is a web service designed as a reusable component in process automation. It exists within the larger context of ESA (enterprise services architecture), and it contains metadata about its functionality and about how it connects to other services. Enterprise services are large enough that combining and recombining them is a fairly easy task. An enterprise service, when called, will execute any number of instructions across any number of underlying applications whereas a web service will call only the application to which it is related (Woods and Mattern, 2006). An enterprise service is composed of the service interface and service implementation. Enterprise services are gateways to functionality provided by an existing system, called a service provider.

Aggregating Web services into business-level enterprise services provides a more meaningful foundation for the task of automating enterprise-scale business scenarios (sap.com). Descriptions of enterprise services are stored in the Enterprise Services Repository, which contains not only WSDL files but also models that show how an enterprise service is related to business processes and business objects. As such, ESA can be defined as “the sum of SOA (Service-oriented architectures) and ES (Enterprise services)” (Feurer, 2007). Combined use of SOA and ES allows developing a reusable library of services having service definitions. As such, combined use of BPM (Business Process Management), SOA, XML (Extensible Mark-up Language) and Web Services enables the formation of a service-oriented enterprise, as given in Figure 1.

In this structure, not all enterprise applications are able to expose the same amount of functionality with the same level of ease. Enterprise services resting on top business objects, an organized container of functionality and data designed specifically to operate well within an ESA framework are able to offer a greater variety of service operations more easily than functionality from an enterprise application that was never designed to provide services (Earl, 2005).

Therefore, enterprise services residing within ESA are loosely coupled, and the composition is not hard coded but rather assembled through process orchestration and modelling. They are just combinations and recombination of underlying services. Therefore, reconfiguring the underlying scenarios, business processes, and process steps become the real issue (Earl, 2005). As such, they can be thought of as standardized Lego pieces to create composite application. These components are kept in the enterprise services repository to allow for reusability. As such, composite applications are essentially applications built out of services provided by other applications. They are constructed using web services as building blocks (Woods and Word, 2004) and created through modelling rather than through a programming language.

A STUDY ON AVAILABLE APPLICATION INTEGRATION FRAMEWORKS

With the critical concepts of SOA and web services already mentioned, an architectural study of various integration platforms from proven vendors is performed. The study focused on the following as representatives of today’s major proprietary integration solutions:

1. Microsoft BizTalk
2. IBM WebSphere
3. SAP NetWeaver
4. Oracle Fusion MiddleWare SOA suite

Based on Chappell (2005), Simmons (2005), Woods and Mattern (2006) and oracle.com, various characteristics, layered architectures, schematic high-level diagrams and main components (together with the functionalities and interactions among them) are analyzed for these platforms, details of which can be obtained from Akyuz (2008). For the sake of conciseness and clarity, these diagrams are not included in this paper. Basing on the study, it can be argued that these structures reveal consistent results and major commonalities. Although there are slight changes in the naming and responsibilities of system components, commonalities observed are striking and the following are worth mentioning:

1. Master data management (MDM): MDM components of the integration platforms provide back-end database integration at both the sender and receiver sides, providing the ability to integrate with ERP backbones.

2. SOA: All of the integration platforms clearly exhibit a service oriented structure, using web services as the enabler of web-based communication. Ability to use standard web services and creating composite applications from available enterprise services are definitely presented via development tools supporting composite application generation. Therefore, it can easily be argued that SOA plays a crucial role in the architectures of most commonly used integration platforms of today.

3. Layered structures: All of the integration platforms mentioned exhibit layered structures. For managing complexity and assignment of functionalities, use of conceptual layers appear as a common principle. One component of an integration platform may be acting in more than one layer, such as SAP XI, entering into action

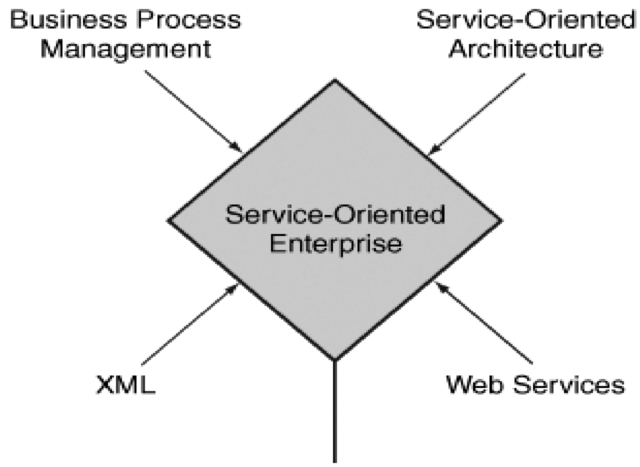

Figure 1. Relationship between ESA and SOA. Source: Woods and Mattern (2006).

for both orchestration and service definitions. Similarly, creation of reusable services involves the use of more than one component and interaction of different components at different layers.

4. BPM: All of the mentioned platforms include components for business process management. This requires providing the definition of business process logic, having a business process rules engine, relating the business rules to the process logic and relating the messages to the business logic. Easy-to-use, graphical tools and user interfaces for defining and maintaining business logic are also provided. Inclusion of business process management capabilities clearly serves the need for convergence of SOA and BPM. Defining and synchronising way of doing business with the technological architecture is provided by incorporating BPM into the integration architecture. This is vital for the success of any implementation, which is evident from the best practice guidelines and discussion of challenges involved in ERP, e-procurement and e-supply chain implementations. As such, any integration platform should be able to handle business process definitions and alignment of business processes with the technology, involving reengineering if necessary. Together with this alignment, this principle ensures the separation between process control and process logic.

5. Business Intelligence (BI): Consolidating data coming from various sources into reports that allow for intelligent managerial decisions is vital for business success and this need is taken care of in all of these architectures. Activity monitoring, alerting, on-line analytical processing (OLAP), data warehousing and drill-down reporting mechanisms are definitely common characteristics.

6. Portals as user interface: Provides the customization of interfaces and separation of user interface from business data and logic. This enables modifying the user interface while keeping the business data and logic intact, making the portal and back-end integrations independent of the sender and receiver platforms respectively.

7. Separation of concerns: Refers to the ability to identify, encapsulate, and manipulate parts of software relevant to a particular concern (concept, goal, purpose). Concerns are the primary motivation for organizing and decomposing software into manageable and comprehensible parts, which reduce software complexity, improve comprehensibility; promote traceability; facilitate reuse, non-invasive adaptation, customization, and evolution; and simplify component integration. Facilitates reuse and evolution of system components or systems as a whole (IBM Research, 2007).

8. Separation of business data and logic: Enables modifying the business rules and logic without changing business data.

9. Scalability: The fundamental infrastructure should be designed to scale up in order to support current message volume and future growth.

10. Extensibility: A good integration solution must be customizable and extensible. A company should be able to add to and change business processes without affecting the underlying applications, and IT should be able to change applications without affecting business processes.

11. Redundancy: Needed to support fault-tolerant configurations in order to be used as part of the mission-critical application solution.

12. Single sign-on: Vital design principle needed for proper authorization and authentication.

All these findings are definitely compatible and supported by LaFata and Hoffman (2004) in terms of key features and functionalities of EAI software suites.

These commonalities are also compatible with Legner and Vogel (2008), emphasizing that service-based interoperability requires agreements and standardizations at pragmatic, semantic, syntactic and communication/ transport levels. Legner and Vogel (2008) clearly indicate that SOA concepts cover only the lower two layers (transport/communication and syntax) leaving the other two layers (pragmatic and semantic) as domain specific and to be addressed by vertical integration. This clearly explains the the need and importance of master data management and business process management in today's EAI platforms.

BENEFITS OF SOA

Literature provides very strong support for the benefits of SOA from various aspects for information systems development and integration.

Woods and Mattern (2006) mention the following in this regard:

1. Greater flexibility.
2. Promoted modularity.
3. Better managing of the complexity.
4. Expanded reuse of existing functionality.
5. Improved communication between IT and business.
6. Simplified and accelerated development and faster time to market through improved developer productivity based on model-driven development, removing IT bottlenecks.
7. Easier adaptation through modeling and role-based tools.
8. Clearly defined roles from the business analysts to the developers.
9. Better encapsulation to allow heterogeneity or outsourcing.
10. Lower TCO (total cost of ownership).
11. A foundation for an ecosystem, which enables the development of an ecosystem of interacting enterprises.
12. A foundation for harvesting value from standards.

Earl (2005) clearly emphasises that by being based on open standards, SOA promotes intrinsic interoperability, federation, architectural composability and reusability. These benefits are definitely in line with ESA principles mentioned by Woods and Mattern (2006) and the key aspects of loose coupling, autonomy, abstraction, reusability, interoperability and composability mentioned by Candido et al. (2009).

Microsoft Whitepaper (2006) argues that by the use of SOA, simplification of management of distributed resources across multiple platforms, reduced hardware requirements, increased reliability and reduced costs are reported, adding upto a dramatic increase in agility and productivity.

Lim and Wen (2003) mention that "web services accomplish better data integration and unlock business information from the inside of information silos". Bolloju and Turban (2007) also emphasizes that for the purposes of systems integration, SOA offers universal communication because of vendor, platform and language independence and flexibility due to loose coupling.

Canfora et al. (2007) consider SOA as a "new chance to continue using and reusing the business functions provided by legacy systems". As such, they mention SOA and web services as "a means of modernizing software systems", emphasizing them as "valuable options for extending the lifetime of mission-critical legacy systems". Exposing legacy systems as services allows heterogeneous systems to become interconnected and interoperable.

SOA is mentioned as "means of attaining greater business agility from existing IT investments, a means to connect systems, workgroups, or geographically distributed subsidiaries or to collaborate with trading partners" (Microsoft, 2006).

Woods and Mattern (2006) emphasises the importance of ESA because of the following reasons:

1. ESA provides a blueprint for all levels of the enterprise architecture, not just for an application but also for "a platform for flexible automation of business processes".
2. ESA provides the ability to "align the business architecture, the application architecture, and the technology architecture".
3. ESA standardises business semantics by providing services that can be used to implement standards and make them useful, or to model and implement relationships among companies.
4. ESA allows the "complexity of applications to be encapsulated in reusable enterprise services that are orchestrated through modeling", allowing development of faster and flexible composite applications possible.

As such, it is evident that by providing flexible, modular, reusable and interoperable infrastructures, the use of SOA provides the ability to cope with the complexities of application integration, including the integration of both existing systems and the new system development. Ability to encapsulate the existing systems, providing the interoperability of totally different platforms and enabling modular system development stand out as the most critical aspects of the SOA.

RELEVANCE OF SOA TO E-SUPPLY CHAİN

The benefits of SOA have been discussed earlier. Here the relevance of SOA from e-supply chain perspective is analysed.

Integration is one of the keys to effective supply chain management (Cheng et al., 2010), helping reduce cost, improving responsiveness to changes, increasing service level, enabling better information sharing, enhancing supply chain visibility and avoiding information delays and distortions. Therefore, as the most current technology for integration, the use of SOA has direct relevance in the dynamic reconfiguration of supply chains, making them readily adaptable to changing business models, growing globalization and increasing coordination. Vernadat (2007) clearly argues that supply chain agility requires interoperability, which is obtained by the use of SOA. As such, SOA becomes an enabler for e-supply chain agility.

Kumar et al. (2007) argue that electronic integration and coordination among supply chain partners leads to more real time information exchange, eliminating the need for replicating the functionality from one system onto another. As such, all these benefits are well suited to be enhanced by SOA, having direct effect on better integration of CRM and SRM systems among partners, enabling collaborative planning and decision making. This is totally suited with e-supply chain definition, which essentially means "the impact that Internet has on the integration of key business processes from end user through original suppliers (Gimenez and Lourenço, 2004). Therefore, systems using SOA can provide functionalities beyond enterprise boundaries and eliminate many shortcomings of ERP systems (Cheng et al., 2010) and makes information sharing across silos easier (Kumar et al., 2007). Thus, presence of SOA would further enhance the benefit by making real time communication and information sharing easier because of SOA's inherent standards based interoperability feature. Hence, a firm with SOA would be in a better position to leverage its electronically integrated customers, suppliers or partners to achieve better electronic supply chain performance (Kumar, 2007) .

Woods and Mattern (2006) also provides support for increased flexibility of supply chain, by arguing that SOA "preserves the gains of the previous generation of enterprise applications", by assuring that all of the standard processes that made ERP, CRM, and other enterprise applications which are vital to efficient operations will stay in place. It is argued that it is not important where these services originate—whether in ERP, CRM, or SCM—since it is now possible to orchestrate them independently via the enterprise services repository incorporating both, a central tank of services that are created for customers and services that companies create on their own.

Discussing the benefits of SOA from the business point of view, Microsoft whitepaper (2006) argues that SOA solutions promote the following, which are again directly related with supply chain efficiencies:

1. Stronger connections with customers and suppliers: By making dynamic applications and business services available to external customers and suppliers, not only is richer collaboration possible, but also customer/partner satisfaction is increased. SOA relieves critical supply and demand chain processes (such as outsourcing of specific business tasks) from the constraints of underlying IT architectures, enabling better alignment of processes with organisational strategy.
2. Enhanced business decision making: By aggregating access to business services and information into a set of dynamic, composite business applications, decision makers gain more accurate and more comprehensive information, together with the flexibility to access that information in the form and presentation that meets their needs.
3. Greater employee productivity: By providing streamlined access to systems and information and enabling business process improvement, businesses can drive greater employee productivity. Employees can focus their energies on addressing the important, value-added processes and on collaborative, semi-structured activities, rather than having to conform to the limitations and restrictions of the underlying IT systems.

Lim and Wen (2003) identified case studies where SOA adoption led to cost savings and increased business efficiency. Andrianopoulos (2002), Murtaza and Shah (2004) and Pereira (2009) also suggest that web services and SOA adoption leads to efficiency in supply chains.

Kumar et al. (2007) also provide empirical support for the impact of SOA adoption on the performance of electronic supply chains for a cross section of large US firms via a survey-based study and conclude that SOA moderates firm's ability to leverage electronically integrated customers to achieve better electronic supply chain performance. As being one of the earliest empirical studies on e-supply chain and SOA relation, they measure performance as "supply chain effectiveness" and uses ordinary least squares regression analysis with robust standard errors to provide evidence for the use of SOA on e-supply chain performance.

Thus, in literature there is a broad agreement that SOA adoption leads to improvements in supply chain performance, having direct positive effect on efficiency, flexibility, agility and the degree of collaboration among supply chain partners. By providing platform-independant communication, SOA frees up supply chain processes from IT constrains.

DISCUSSION

The benefits and importance of SOA from application integration and e-supply chain perspectives had been revealed. It can be easily argued that these offerings are definitely in line with e-supply chain formation requirements and challenges classified and discussed by Akyuz and Rehan (2009).

Akyuz and Rehan (2009) discuss e-supply chain requirements and challenges from both technological and business perspectives. They mention the following as the main requirements for forming an e-supply chain:

1. Replacement of or integration with the legacy systems.
2. Standardising and streamlining internal processes: BPR (business process reengineering)/ redesign if needed.
3. Adoption, updating or integrating with the existing ERP of the enterprise.
4. Standardising and streamlining external processes-BPR/redesign if needed.
5. Collaborative planning and joint management of key business processes.
6. Business intelligence and decision support.

It is evident that these requirements are compatible with Microsoft whitepaper (2006) classification of SOA benefits from business viewpoint. "Stronger connections with customers and suppliers" is definitely what is needed for e-supply chain formation. "Enhanced business decision making" is a giant step towards web-based collaboration, which is one of the critical requirements of e-supply chain formation as mentioned in Akyuz and Rehan (2009). "Greater employee productivity" is again a great contribution to e-supply chain efficiencies.

Consideration of legacy system integration among the critical requirements of e-supply chain formation is again compatible with the ideas mentioned in Canfora et al. (2007) and Woods and Mattern (2006). As such, SOA provides a critical support for legacy systems integration, preserving the gains of previous systems and helping in extending the lifetime of mission-critical systems.

Integration of business process management tools into SOA architectures definitely serves business process standardization requirements of e-supply chain and similarly integration of business intelligence tools serve for collaborative planning and joint management of key business processes. Therefore, e-supply chain requirements put forward in Akyuz and Rehan (2009) appear to be compatible with today's SOA-based application integration platforms and SOA benefits discussed in literature.

Among the technological challenges involved in e-supply chain formation, Akyuz and Rehan (2009) mention that "supporting on-line, real-time connectivity and visibility among heterogeneous, possibly inconsistent information systems in a secure, reliable and efficient manner" is still the main issue in the e-supply domain. As such, the highest degree of interoperability, platform independency, standardisation and loose coupling offered by web services "discussed and supported previously appears to be exactly what is needed for this main issue.

It is also made clear that encapsulation of legacy systems, obtaining platform independence, preserving the gains of the previous generation of enterprise applications, aligning the business processes, application and the technology architecture are definitely desired properties for e-supply chain formation.

All these ideas are supported by (Bolloju and Turban, 2007), reporting that early adopters have employed web services for the following types of systems integration:

1. Integrating legacy systems with other systems by wrapping existing functionalities as web services.
2. Consolidating data extracted from different applications and databases using Web services and providing that data through web services
3. Accessing different applications through web services from multiple interfaces
4. Enhancing existing applications using external web services

As such, web services currently represent the most proper technology for the ultimate integration in the form of e-supply chain.

Basing on all these discussions, it can be easily argued that "SOA acts as the critical enabler to meet the requirements of e-supply chain formation and application integration, serving the needs of legacy systems encapsulation, better external integration, joint planning and collaboration, business intelligence and decision support needs". This idea is clearly supported by Bih (2006) who names SOA as "the new paradigm to implement dynamic e-business solutions".

Transition to SOA is a radical shift from traditional application development, with the focus of model-driven orchestration of enterprise services, leading to the formation of composite applications development (Gartner Reports; Woods and Word, 2006; Earl 2005). As ESA continues taking shape, it appears that the basic structure of the future will be the business object, which is a container of functionality where data will be managed and processed. These business objects will then be grouped together into larger containers, on top of which will rest enterprise services that allow for external access to the business objects beneath. The result is that enterprise applications will no longer be user interfaces to monolithic functionality and instead will become user interfaces resting on top of process components composed of related sets of business objects exposed for external use as enterprise services (Güner, 2005). The authors hold the opinion that these principles will definitely characterise the e-supply chain infrastructure.

To emphasise the importance of SOA and web services, Woods and Word (2006) make the following quote from a Gartner Group Report:

"There is no alternative to [SOA] and web services as a

basis for future software. The issues revolve around the rate of adoption and the purposes for which it is applied. In other words, it is not a question of whether an SOA will supplant today's architecture, but rather, how long it will take to complete this evolution ".

Keeping all these in mind, it can easily be argued that any infrastructure used for e-supply chain formation should be compatible with SOA design principles to better serve for the requirements of e-supply chain formation. But still, it should be kept in mind that integration levels needed for e-supply chain formation requires much more than use of SOA. As Legner and Vogel (2008) argued, solution to pragmatic and semantic integration still lies in business process management and master data management.

CONCLUSION

This article put forward the rise of SOA trend in application integration by discussing evolution, critical concepts and integration platform structures from four main proprietary vendors. Commonalities and similarities in these four platforms turned out to be striking, revealing a clear shift towards SOA and showing that proven software vendors have currently reached consistent architectures. Strong backend integration with ERP systems, use of web services, portal-based user interface, composite application development tools, system management tools and business intelligence support overlaid with this basic infrastructure appear as the basic characteristics of application integration.

Benefits of SOA and its relevance to e-supply chain formation are discussed in detail and it is made evident that use of SOA is definitely in line with e-supply chain formation requirements. As such, it appears that SOA is still the current answer and the greatest enabler for assuring platform independent application integration, which is the basic challenge of e-supply chain formation and ultimate web-based collaboration.

Regarding the issue, organisational assimilation of the technology is still an open research area as suggested by (Bolloju and Turban, 2007). Case-based surveys will be significant contribution in this area.

REFERENCES

Akyuz GA (2008). A Survey Based Study on the role of e-procurement in integrating ERP (Enterprise Resource Planning) systems using e-supply chain. Thesis (Ms in Computer Engineering). Atilim University.

Akyuz GA, Rehan M (2009). Requirements for forming an 'e-supply chain'. Int. J. Prod. Res. 47 (12): 3265–3287.

Andrianopoulos A (2002). The framework behind Webservices integration. EAI J., June, 22-23.

Bih J (2006). Service Oriented SOA: A new paradigm to implement dynamic e-business solutions, whitepaper [online]. Avaiable from: http://whitepapers.zdnet.co.uk/0,1000000651, 260276228p,00.htm. [Accessed 15 October 2007].

Bolloju N, Turban E (2007). Organizational assimilation of web services technology: A research framework. J. Organizational Computing Electronic Comm. 17(1): 29-52.

Candido G (2009). SOA in reconfigurable supply chains: A research roadmap. Eng. Appl. Artif. Intell., 22 (6): 939–949.

Canfora G. et. al (2008). A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures. J. Syst. Software, 81(4):463-480.

Chappel D (2005). Microsoft Corporation. Understanding BizTalk Server 2006. White paper [online]. Available from: http://download. microsoft.com/documents/australia/windowsserversystem/biztalk200 6/Understanding_BTS06.pdf [Accessed 15 September 2007].

Chenga JCP, Lawa KH, Bjornssonb H, Jonesc A, Sriramc R (2010). A service oriented framework for construction supply chain integration. Automation in Construction 19 (2): 245–260.

Ciol.com (2002). Integration levels in EAI. [online]. Available from:http://www.ciol.com/content/search/showarticle1 .asp?artid=40712 [Accessed 15 November 2007].

Earl T (2005). Service Oriented Architectures: Concepts, Technology and Design [online]. Available from: http://acmsel.safaribooks online.com/0131858580. [Accessed at various dates].

Feurer S (2007). Enterprise Architecture: IT Meets Business [online]. Available from:http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs /library/uuid/403fd993-56d9-2910-eaa4-9c52bcd1b82d [Accessed 11 December 2007].

Gartner Group Research (2007). Evaluating IBM, Microsoft, Oracle and SAP commitment to SOA governance, Research Note: G00150869 [online]. Available from:http://www.sap.com/platform/esoa/index.epx [Accessed 15 November 2007].

Giménez, C, Lourenço, HR (2004). E-Supply Chain Management: Review, implications and directions for future research, Institut D' estudis Territorrials, Working Paper no: 17.

Güner S (2005). Architectural approaches, concepts and methodologies of SOA, Thesis (Ms in Software System Institute), Technical University Hamburg, Germany. Available from: http://wwwsts.tu- Available from: harburg.de/pw-and-m-theses/2005/gune05.pdf [Accessed 15 October 2007].

HimalayanTechies (2004). Enterprise Application Integration [online]. Available from: http://www.himalayantechies.com/enterpriseDev.htm [Accessed 9 October 2007].

Hündling J, Weske M (2003). Web Services: Foundation and composition. Electronic Markets 13(2):108-119

IBM (2007). Application integration [online]. Available from: http://www.ibm.com/developerworks/patterns/ application/ [Accessed 1 November 2007].

IBM Research 2007. Multi-dimenional Separation of concerns: Software Engineering Using HyperSpaces [online]. Available from: http://www.research.ibm.com/hyperspace/ [Accessed 25 October 2007].

Kumar S et al (2007). Does SOA Improve the Supply Chain? An Empirical Analysis of the Impact of SOA Adoption on Electronic Supply Chain Performance. IEEE, Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07).

Kunti K *et al.* (2007). SOA Based Integration for creation of collaborative supply chain in automotive domain. SETLabs Briefings.5(3), July-Sept.

LaFata J, Hoffman S. (2004). Enterprise Application Integration: A primer in Integration Technologies [online].Available from: http://www.liquidhub.com/docs/Horizons_EAI.pdf [Accessed 18 July 2008].

Legner C, Vogel T (2008). Leveraging Web Services for implementing vertical industry standards: a model for service-based interoperability, Electronic Mark. 18(1):39-52.

Lim B, Wen JH (2003). Web services: An Analysis of the technology,its benefits and implementation. Inf. Syst. Manag., 20(2):49-57.

Lior F, Seev N (2009). Taking the high road to web services implementation: An exploratory investigation of the organizational impacts. ACM SIGMIS Database. 40(3): 84-108.

Microsoft (2006). Learn about Service Oriented Architecture (SOA) [online]. Available from: http://www.microsoft.com/biztalk/solutions/ soa/overview.mspx [Accessed 3 November 2007]

Murtaza, MB, Shah JR (2004). Managing information for effective business partner relationships. Inf. Syst. Manage., 21 (2): 43-52.

Nickull D (2005). Service Oriented Architecture, Technical Whitepaper [online]. Available from: http://www.adobe.com/enterprise/pdfs/ Services_Oriented_Architecture_from_Adobe.pdf. [Accessed 3 November 2007]

OASIS (2005). Web Service Implementation Methodology [online]. Availablefrom: http://www.oasis-open.org/committees/download.php/ 13420/fwsi-im-1.0-guidlines-doc-wd-publicReviewDraft.htm#_Toc105 485354 [Accessed 12 November 2007]

ORACLE (2007). Accelerate your Enterprise SOA by leveraging the application integration architecture Foundation pack, whitepaper [online]. Available from: http://www.oracle.com/applications/aia-foundation-pack-white-paper.pdf [Accessed 12 October 2007]

ORACLE (2007). Oracle SOA Suite [online]. Available from: http://www.oracle.com/technologies/soa/soa-suite.html. [Accessed 10 December 2007].

Pan A (1999). Enterprise Application integration- message broker style, Sun World [online]. Available from http://sunsite.uakom.sk/sunworldonline/swol-08-1999/swol-08-itarchitect.html. [Accessed 28 September 2007].

Papazoglou M, Heuvel, WJAM (2007). Service oriented architectures: Approaches, technologies and research issues. Very Large Database J., 16(3): 389-415.

Pereira JV (2009). The new supply chain's frontier: Information management. Int. J. Inf. Manage. 29(5): 372–379.

SAP NetWeaver [online]. Available from: https:www.sdn.sap.com/ irj/sdn/developerareas/netweaver [Accessed 9 March 2007].

SAP (2007). Service oriented architectures [online]. Available from: http://www.sap.com/platform/esoa/index.epx [Accessed 08 September 2007].

Simmons S (2005). Introducing the WebSphere Integration Reference Architecture: Introducing a service based foundation for enterprise-level business integration, IBM WebSphere Developer Technical J., [online]. Available from: http://www.ibm.com/developerworks/websphere/techjournal/0508_simmons/0508_simmons.html [Accessed 10 October 2007].

Sweat J (1999). The integrated Enterprise [online]. Available on-line: http://www.informationweek.com/731/eai.htm . [Accessed 10 October 2007].

Vernadat FB (2007). Interoperable enterprise systems: Principles, concepts, and methods. Ann. Rev. Control 31(1):137–145.

Woods D, Mattern N (2006). Enterprise SOA: Designing IT for business innovation. O'Reilly [online]. Available from http://acmsel.safaribooksonline.com/0596102380 [Accessed at various dates for October and November 2007].

Woods D, Word J (2004). SAP NetWeaver for Dummies, Canada, Wiley Publishing Inc.