

Business Interactions with Web Services

Youcef Baghdadi

Department of Computer Science, Sultan Qaboos University, PO Box 36, PC 123 Al-Khod, Oman
ybaghdadi@squ.edu.om

ABSTRACT

Business interactions require technologies with attractive features to dynamically connecting enterprise information systems and external applications. This work presents a framework of business interactions that consists of: (1) a conceptual architecture view, (2) a technical architecture view, and (3) an implementation architecture view of business interactions. The conceptual view defines and categorizes business interactions into enterprise interactions, partners' interactions and customers' interactions. The categorization allows deciding and deploying the technical view that specifies the required connections for EAI, CRM, B2C and B2B. The implementation view shows how the features of Web services, a connecting technology, enables any kind of specified business interactions. Web services features allow first unlocking and adding value to existing enterprise assets such data and applications; then dynamically connecting them into new solutions such as business processes that respond to business events or changes in business conditions.

Key Words: Business Interactions, Framework, EAI, CRM, B2C, B2B, e-Business, Integration, Web Services, SOA.

1. INTRODUCTION

Businesses need to adapt rapidly and cost-effectively to new businesses events and conditions, namely customers and partners demand and behavior. This requires technologies that enable cost-effective and dynamic interactions within and especially across the boundaries of the enterprise. However, to deploy any technology, one must abstract the business according to different but related perspectives such as in [16]. This work concerns with a business interactions perspective, namely a framework with three views of business interactions in order to decide and deploy a technology with attractive features, namely connecting cost-effectively existing assets in terms of data and applications.

The framework aims mainly at providing business interactions oriented architecture that considers both business perspective and IT perspective in order to point up design as well as technical issues that may hinder the deployment of business interactions enabling technology. That is, Web services.

The framework consists of three architectural views of business interactions perspective that are:

1) The conceptual view architecture aims at specifying the business interactions regardless of the implementing technology. The conceptual view architecture defines, specifies and categorizes business interactions into customers' interactions, enterprise interactions and partners' interactions. This categorization allows a specification of each category of interactions in order to decide an appropriate enabler technology for respectively EAI, CRM, B2C, B2B, and ultimately dynamic e-business.

2) The technical view architecture aims at identifying: (i) the existing enterprise informational as well as computational assets (e.g. existing applications, integration applications and middleware), (ii) the existing (or to develop) connections between the enterprise assets and external applications, and (iii) the requirements and features of the connecting technology. The technical view focuses on technological requirements to fill the business processes breakups that require connections.

3) The implementation view architecture specifies how Web services technology, the *de facto* integration standard [1]; and also the *de facto* Internet standard instance of the services-oriented architecture (SOA) [2][14], presents attractive features in terms of cost-effective connections within and across the enterprise, i.e. between the enterprise and its customers/partners to enable EAI, CRM, B2C, B2B, and ultimately dynamic e-business. Web services technology is, by essence, a connections technology. Indeed, the underlying standards of such a technology, namely, XML, WSDL, UDDI, WS-I, BPEL4WS, SOAP and other related to security and transactions when matured will allow: (1) interfacing the enterprise assets that are data and applications as services in order to unlock them, (2) publishing the interfaces in a registry to be discovered and reused, (3) discovering the services, (4) connecting (statically or dynamically) any application to the services, (5) ultimately connecting services to each other with respect to SOA architecture, and (6) composing dynamically the services into new solution to respond to business events or changes.

2. CATEGORIZATION OF INTERACTIONS

Due to organizational and technical constraints, business processes are broken up into causal and reciprocal activities. That is, the set of activities form a

flow where each activity takes an input from a provider and serves an output to a consumer. These business processes breakups require a set of business interactions, which are generally implemented by organizational and technological artifacts. Therefore, one needs to abstract these interactions in order to first specify them with respect to a business perspective before to come to a technology perspective. The abstraction allows deciding an appropriate technology that enable cost-effective and dynamic interactions.

2.1 Business Interactions

The concept of interactions has been specified by different disciplines. A business perspective specification, we adapt from different disciplines, considers interactions as a set of reciprocal (or causal) activities performed by local or remote providers and consumers in a synchronous or asynchronous manner. Each of them has a certain degree of autonomy and freedom (e.g. to leave at any time). Therefore, business interactions are characterized by space, time, and dynamics. Moreover, providers and consumers interact in different situations.

Space. Interactions involve local as well as distributed and remote providers and consumers.

Time. Action-reactions of the providers and consumers are temporal, synchronous as well as asynchronous.

Dynamics. While interactions in goal-driven businesses are often rigid and fixed (e.g. mechanisms of coordination used to offset planned task allocation in goal-driven businesses). Interactions, in event-driven businesses, are dynamic relationships among autonomous providers and consumers.

Situations. Situations of interactions are answer to 'why do providers and consumers interact?' With respect to a business (and information systems) perspective. Business providers and consumers interact for numerous reasons, namely:

- Internal and external exchanges of information.
- Synergy of business processes.
- Unlocking informational (e.g. data) and computational resources (e.g. applications).
- Emergent knowledge. Indeed, knowledge resulting from interactions is more relevant and more complete than simple knowledge integration.

2.2 Categorization of Business Interactions

In addition to their specification with respect to space, time, dynamics, and situations, it is a must to categorize business interactions while deciding and deploying enabler technologies. First, business interactions differ according to the types of the involved providers/consumers, and the types of business processes and business events. Indeed, local

business interactions are different from interactions with partners and customers. Moreover, the business interactions of a goal-driven business differ from those of an event-driven business. Second, business interactions will not have the same priority with respect to the enabling technology. For, instance, EDI technology may be suitable for a long term fixed relationship between a business and its partners. Whereas Web services technology allows a real dynamic relationship between a business and its partners. Third, interactions between a business and its partners/customers are based on the effectiveness of internal interactions. That is, one cannot design and implement external interactions if the local ones are not working properly. For instance, B2B cannot be approached if EAI is not working properly. Figure 1 distinguishes each of the following categories of business interactions:

- 1) Interactions between business and its customers (C-B).
- 2) Internal (enterprise) interactions that are:
 - Interactions between enterprise (local) primary processes (P-P).
 - Interactions between local primary and supporting processes (P-S).
 - Interactions between local business process activities (AT). These are required when a local business process is decomposed (A-A).
 - Interactions between business processes and the business objects (or coordination artifacts) as presented in the different information systems and legacy systems.
- 3) Interactions across businesses, i.e. interactions between business processes that cross the boundaries of the organization (B-P).

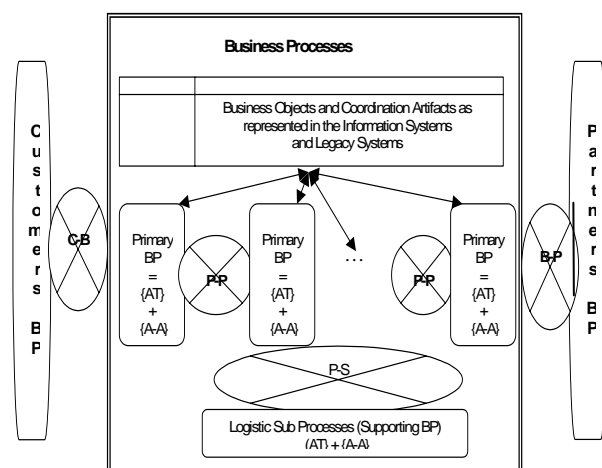


Fig. 1. Categories of Business Interactions

2.3 Interactions Common Activities

Regardless of their characteristics, i.e. dynamics, time, space, situations and categories, interactions require announced partners to perform the following activities as shown in Figure 2: (a) Identify the partner, (b) Send a message to the partner, (c) Read and interpret the message, (d) Understand the message, (e) Act

according to the interpretation and understanding (semantics required), (f) Receive a message, (g) Read and interpret the message, (h) Understand the message, and (i) Re-act according to the interpretation and understanding (semantics required).

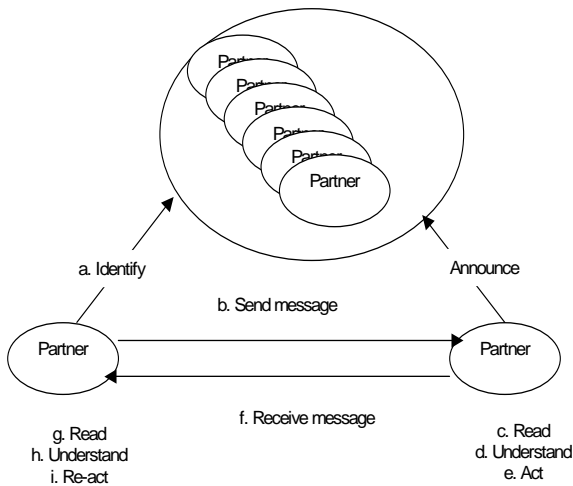


Fig. 2. Interactions Common Activities

3. TECHNICAL ARCHITECTURE VIEW

The technical architecture view (Fig. 3) aims mainly at specifying the whole business IT system as a set of connecting IT subsystems that support business processes. The IT subsystems may be any combination of information systems, legacy systems and other kinds of applications. The interacting IT subsystems are:

1. Local subsystems that are different enterprise information systems (EIS) and legacy systems (LS) used to sustain primary and supporting business processes.
2. Clients subsystems that interact with local IT subsystems, these are stand-alone clients (SC: clients that interact through standalone applications), Web browsers (BC: clients that interact through browsers) and mobile clients (MC: clients that interact through mobiles).
3. Partners subsystems that interact with local IT subsystems, these are the partners information systems, legacy systems, components, objects or services.

The connecting IT subsystems are interfaced and viewed as endpoints. That is, their internal behavior is encapsulated as a set of endpoints. The interfaces depend on the types of IT subsystems. That is, legacy systems and to a less extent enterprise information systems or applications will be wrapped to expose public interface. While, the components, objects and services are, by essence, specified as interfaces and implementations. The interfaces are:

1. Interfaces of the IT subsystems that are implemented with enterprise applications integration (EAI)

according to the integration orientation (data, applications or business processes) and different types of middleware (transactional, remote procedure call, messaging, message brokers, distributed objects computing, or SOAP).

2. Interfaces of the partners IT subsystems that are implemented through B2B electronic commerce applications (e.g. EDI, ebXML, Web services) or e-market brokers.

3. Interfaces of the clients that are implemented through B2C and CRM applications.

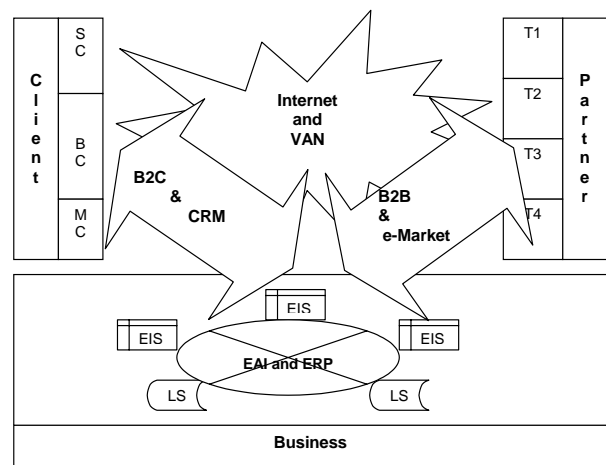


Fig. 3. Technical View of Business Interactions

This connection-oriented technical architecture view represents different categories of business interactions. To effectively allow dynamic business transaction, the implementing technology must emphasize connection of transparent and self-contained existing enterprise assets that don't tightly depend on each other. That is, a technology that allows loose coupling of the interacting elements. Web services technology is the *de facto* connections technology when deployed with respect to SOA architecture.

4. IMPLEMENTATION WITH WEB SERVICES

The technical architecture view is implemented with different technologies. Indeed, business interactions have been implemented as EAI, ERP, B2C, CRM, and B2B using traditional middleware, and distributed object-computing middleware (e.g. CORBA, DCOM and RMI) before the advent of SOA architecture and its instances, namely e-services and Web services. SOA architecture with Web services technology, despite the lack of maturity of the underlying standards, presents advantages with regard to (1) loose coupling of interacting elements, (2) dynamics, and especially (3) the connection-oriented applications of Web services.

4.1 SOA Architecture

SOA architecture was first described by Gartner Group in 1996 as architecture based on the concept of service. It is essentially a collection of services that communicate with each other [2]. There exist several definitions of the term service. For instances, a kind of business oriented definition is given by [10] who defines a service as “business function of an application, so that another application or an application at another enterprise may find it useful to invoke”. A more technical definition is given by [13] who defines a service as “a location-transparent, network-addressable, invocable unit of software logic that is well defined, self-contained, and does not dependent on the context or state of other services”.

We consider a service as a self-contained element of the information system that implements business logic, business rule, or data operation (insert, update, delete, or retrieve data); and provided with a well-defined and standardized interface to be transparently accessible. This element of the information system must be designed and deployed with respect to SOA architecture. SOA aims mainly at achieving communication among loose coupling interacting software agents [8]. Therefore, Web services technology is the most attractive due to its underlying standards that enable communication between services within and outside the enterprise through a public network that is Internet.

4.2 Web Services as Internet Instance of SOA

W3C/WS Architecture Group defines Web services as “software system identified by URI, whose public interface and bindings are defined and described using XML. Other software can discover its definition. These software may then interact in a manner described by its definition using XML-based messages conveyed by Internet protocols” [1].

Web services technology is mainly a collection of technologies and standards that allow connecting services over the Web. These technologies and standards allow interfacing, publishing, binding, composing services through communications protocols based on Internet protocols (e.g. HTTP, SMTP, MIME). The standards XML, WSDL, UDDI, WS-I, BPEL, and SOAP allow communication and execution via the Web of self-contained and loosely coupled services within and outside the enterprise, which makes Web services technology not only the de facto integration standard, but also the de facto Internet standard instance of SOA architecture [14]. Web services technology can easily live with other technologies such as CORBA, DCOM, or RMI. These features of Web services will enable all the categories of business interactions as shown in the next section.

4.3 Web Services Enabler of Business Interactions

The above interactions common activities can be easily implemented by the standards and technologies underlying Web services. Figure 4 shows how Web services stack and technologies (bolded in the figure) implement the interaction common activities.

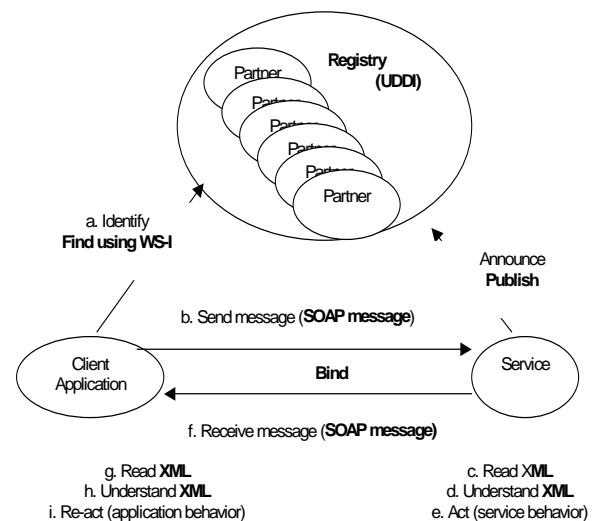


Fig. 4. Web Services and Common Activities

4.4 Web Services Enabler of EAI, B2B, CRM, B2C, and Dynamic e-Business

The features of Web services technology, notably their communication features enable all the categories of business interactions as summarized in Table 1.

4.4.1 Web Services Enabler of EAI

EAI is an integration technology. It consists of providing interface to technologies implementing heterogeneous enterprise information systems and legacy systems. It aims at making applications composite to rapidly respond to business events, and to adapt to change in business conditions. The main features of EAI are:

- Connection of the applications (in different enterprise information systems and legacy systems).
- Introspection to look up and find applications or databases in order to connect them.
- Translation of data and messages exchanged between the applications and databases of the connected systems
- Control of flow of the applications, from within the different connected systems, involved in the composition of business processes.

These features can easily be enhanced with Web services technology. Indeed, Web services technology present transparent view of business logic, rules and data by interfacing and publishing them to be accessed, invoked, and consumed by applications that enter in the composition of primary and supporting business process.

EAI will easily support Web services. Indeed, Web

services are just a step in the evolution of technology and standards. The standards underlying Web services will ultimately simplify application integration within and outside organizational boundaries. These standards (when matured enough) will allow Web services to become the de facto integration standard by supporting all features of EAI. Moreover, Web services reduce the complexity and the cost of the traditional integration middleware.

4.4.2 Web Services Enabler of B2B

A B2B application consists of connecting applications from different businesses to exchange agreed upon (or standardized) business documents (e.g. order, invoice). For large organizations, this has been achieved based on EDI specifications. Despite the evolution of EDI, from EDI data transmission, VAN (used to simplify complexity of multiple connections), DCOM and CORBA (used to reduce the cost related to VAN and to standardize the message), to XML as EDI standards, there are still challenges that can be addressed by XML and Web services. Indeed, Web services (with matured underlying technologies and standards) are used in B2B to:

- Reduce the cost of entry into B2B for small and medium businesses. Indeed, EDI deployment and maintenance is very costly, which deprives small and to a less extent medium businesses to play the right role they are intended to in the economics arena.
- Allow effective dynamics. Indeed, EDI specifications assume a long term fixed and well-specified interactions (e.g. agreed upon format) deployed through proprietary networks. Web services connecting technology is based on the contract of type "take or leave", which gives more freedom and dynamics to businesses to choose transparent services accessed through the protocols of the Internet (a public network).
- Fix the problem of the difference in semantics, and the problem of fixed record format of exchanged between businesses applications as imposed by EDI specifications.

4.4.3 Web Services Enabler of CRM

CRM system is an information system that records information such as customer contact information. A CRM system is generally used by customers and representatives. A CRM system is not necessary an internal system. On the contrary, external CRM systems are better suited to be accessed as Web services. Indeed, CRM as Web services will certainly present better services, performance and reliability.

4.4.4 Web Services Enabler of B2C

Web services will facilitate the development and deployment of applications that can be readily accessed by PDA, office devices, and mobile devices in addition

to Web browsers. Web services will permit applications developers to better leveraging existing businesses logic, rules and data by invoking them by any kind of clients. Moreover, Web services will reduce search complexity for exiting services.

4.4.5 Web Services Add-Value

Web services features add value to traditional applications used to implement internal and external business interactions as summarized in Table 1.

Table 1. Value added by Web services

Interactions Applications	Value added by Web Services Technology
EAI	<ul style="list-style-type: none"> • Standardized interfaces of the elements of the enterprise information systems and legacy systems. • Connection of loosely coupled elements of the enterprise information systems and legacy systems. • Introspections through UDDI and WSDL. • Use of XML to exchange and translate messages. • Control of the flow of the integrated applications through BPEL.
B2B	<ul style="list-style-type: none"> • Entry into B2B to small and medium enterprise (VAN is no longer required). • Dynamic business interactions. Businesses are really autonomous in their interactions • Free message format using XML. Messages format not fixed and standardized.
CRM	<ul style="list-style-type: none"> • Use of external as well as internal CRM applications accessible from anywhere through Internet.
B2C	<ul style="list-style-type: none"> • Applications involved in B2C are better interfaced (WSDL) to be accessible through different means, i.e. office devices, PDA and mobile devices in addition to Web browsers. • Search and tracking facilities exposed as Web services.
Dynamic e-Business	<ul style="list-style-type: none"> • Dynamic binding facility allows business applications or services to bind to services at run-time. • UDDI and WS-I allows inspection and selection of efficient services and cost-effective connection.

5. RELATED WORK

Business interactions have been implicitly seen as an integration problem with an IT perspective rather than a business perspective. That is, the issue is often how to make information systems (including applications and databases), running on different platforms, interoperable to exchange structured as well as unstructured data? Various integration ways have been approached, namely: (1) a data-oriented integration (e.g. distributed databases [3]), (2) an object-oriented integration based on distributed object computing middleware (e.g. COBRA, DCOM, RMI), and (3) a

service-oriented integration based on SOAP middleware (e.g. e-services, Web service).

The object-oriented integration approach makes applications interoperable through a broker used for discovery and invocation. In this category, CORBA, DCOM and RMI were designed mostly from an IT perspective in order to help developers operate more effectively [15]. Due to their great returns and quality productivity, these technologies have been used in EAI, and B2B to offset the complexity related to value added networks (VAN) used as brokers in EDI. However, they do not allow a loose coupling because they are built on their own technology.

In the last years, clear trend is to move away from data-oriented and object-oriented integration to service-oriented integration [2][5][7][8][9][15].

The state of art shows SOA and Web services with praise. This technology is now widely adopted because it allows connecting partners (e.g. B2B integration) with reduced cost [4]. The ultimate goal of this technology is to enable dynamic e-business. Web services technology is a main focus of various software development companies [6]. It is considered as the hottest topic by software industry.

Our approach presents a model of business interactions as a framework that allows guidance towards a method to deploy Web services as add value for the exiting applications that are EAI, B2C, CRM, and B2B. The framework is used to abstract and then deploy a comprehensive and multipurpose set of services with respect to SOA. A model of business interactions simplifies the vision and the technological architecture.

6. CONCLUSION

We have proposed a framework for business interactions, as one of the most important perspectives of a business modeling besides the business processes, business objects and business events perspective. An interactions perspective is critical while deciding the alignment and deployment of an enabling connecting technology. This abstraction is made up of three architectural views of business interactions: conceptual, technical, and Web services.

The main goal of the business interactions abstraction is to propose recommendations and guidance that help generating, and deploying a set of Web services with respect to SAO architecture. The abstraction makes clear the breakups of the business processes and the technological requirements to fill these breakups.

We have shown how the features of Web services technology as Internet instance of SOA architecture add value to the features of EAI, B2C, CRM, B2B in order to ultimately allow dynamic e-business.

Deploying Web services technology with respect to architecture is a critical issue nowadays where businesses need to cost-effectively and dynamically integrate business processes that cross their boundaries, which is critical for their survival.

This work can be extended by developing models and supporting tools towards a method for deploying Web services as an Internet standard instance of SOA.

REFERENCES

- [1] Austin D et al., "Web Services Architecture Requirements", W3C Working Group Draft 14, Retrieved from <http://www.w3.org/TR/wsa-req#id2604831>, 2002.
- [2] Barry D K. , "Web services and Service-Oriented Architecture", Elsevier (USA). 2003.
- [3] Batini C, Lenzirini M, Navathe S. B., "A Comprehensive Analysis of Methodologies for Database Schemas", *Integration. ACM Computing Surveys*, 18(4): pp. 322-364, 1986.
- [4] Box B., "Web Services Revolution", Retrieved http://www.eds.com/thought/thought_leadership_web_revolution.pdf, 2003.
- [5] Cohen L. D., "Building and Managing Adaptive e-Business Solution Infrastructure", Keynote in proceedings of IEEE CEC03, 2003.
- [6] Fensel D, Bussler C., "The Web Service Modeling Framework WSMF", *Electronic Commerce Research and Applications* 1(1): pp. 113-137, 2002.
- [7] Frankel D, Parodi J., "Using Model-Driven Architecture to Develop Web Services", Retrieved <http://www.omg.org/attachments/pdf/WSMDA.pdf>, 2002.
- [8] He H., "What is Service-Oriented Architecture?", In: O'Reilly Open Source Convention: July, pp. 26-30, 2003.
- [9] Linthicum D. S., "Moving to Service-Based Application Integration", *eAI Journal*, October: pp. 9-10, 2001.
- [10] Natis Y., "Service Oriented Architecture Ushers in the Next Era in Business Software Engineering", *BIJ*, May: pp. 23-25, 2004.
- [11] Plesko D and Lee J. M., "Web Services and Enterprise Integrations: Friends, Not Foes", *BIJ*, October, pp. 8-12, 2003.
- [12] Phillips J and Foody D., "Building Foundation for Web Services", *eAI journal*, March, pp. 8-13, 2002.
- [13] Roth P. , "To a Service-Based Architecture", *BIJ*, October: pp. 48-50, 2003.
- [14] Smith D. Web Services Enable Service-Oriented and Event-Driven Architecture, *BIJ*, May: pp. 12-14, 2004.
- [15] Wong L. , "Services: A Key Component for Success", *eAI Journal*, March: pp. 18-25, 2001.
- [16] Zackman J. A., "Concepts of the Framework for Enterprise Architecture", Zackman International Los Angeles, CA, 1996.