Preprint version of chapter published in: Pallab Saha (ed.), *Handbook of Enterprise Systems Architecture in Practice*, pp. 419-433, Information Science Reference, IGI Global, March 2007

Business Networking with Web Services: supporting the full life cycle of business collaborations

Diogo R. Ferreira
IST – Technical University of Lisbon
diogo.ferreira@tagus.ist.utl.pt

KEYWORDS
Business networking, Web services, B2B conversations, life cycle, integration

ABSTRACT
This chapter describes how the latest advances in Web Services technology are paving the way towards dynamic B2B integration. It begins by distinguishing between three kinds of life cycle: the operational life cycle, the B2B trading life cycle, and the business networking life cycle. In the past, most B2B integration solutions have been designed to support the operational life cycle by specifying a set of pre-defined B2B conversations. We will show that Web Services can be used to settle those conversations at run-time, and that they can support the B2B trading life cycle all the way from partner search to contracting, operation and evaluation. In general, though, the B2B trading life cycle may change across different markets, which requires mechanisms for a company to retrieve information about how it should proceed once it enters a new market. These mechanisms belong to the business networking life cycle, and they can be supported by Web Services as well. The conclusion is that Web Services provide the required features to support the full life cycle of business collaborations.

INTRODUCTION

The Web has an unparalleled potential to reshape the way companies conduct their businesses. As a globally connected and widely accessible network infrastructure, the Web allows an enterprise to find and establish business relationships with new business partners. At the same time, it allows an enterprise to integrate its systems and processes with those of its business partners. Provided with a common, global network infrastructure, enterprises can associate with each other in order to become more competitive or to offer improved products or services. The result is the development of business networks, which combine the competencies of several business partners.

Solutions such as EDI, sophisticated websites, e-marketplaces, e-procurement systems (Albrecht, 2005) and B2B frameworks (Shim, 2000) have been at least partly successful in achieving B2B integration, by providing either the infrastructure or the data formats for message exchange. The challenge today is to come up with a technological solution by means of which an enterprise could search for, evaluate, select potential business partners and interact with them in a mostly automated way. We will refer to this kind of dynamic B2B integration as *business networking* (Österle, 2001). We will show in this chapter that Web Services, more than any previous technology, display an unprecedented potential to support business networking.

WEB SERVICES AS AN INTEGRATION TECHNOLOGY

Before the arrival of Web Services technology, integration was about defining interfaces and making sure that systems correctly implemented or invoked those special-purpose interfaces. There was RPC, there was message-oriented middleware, there were transaction-processing monitors, and there was CORBA. Then Web Services came along, and with it came the capability of discovering and binding to interfaces either at build-time or at run-time. Not that the concept is entirely new – CORBA, for example, included a mechanism called Dynamic Invocation Interface (Vinoski, 1997) – but the ability to describe and deploy components that can be easily discovered and invoked, possibly in an automated way, came only with the adoption of WSDL (Chinnici, 2005) and UDDI (Clement, 2004) standards.

As the foundation of Web Services technology matures, it becomes clearer how they will be able to achieve the ultimate goal of dynamic B2B integration. From enterprise application integration to enterprise-wide service-oriented architectures, and then to supporting B2B interactions between different organizations, Web Services technology is a cross-level integration paradigm that allows an application to invoke another component, as well as a company to interact with another organization. At first, the single key enabling feature to these scenarios seemed to be the possibility of publishing, searching for, and discovering Web Services. Today, the problem of integration is known to require other features such as service composition, orchestration, and coordination, besides addressing, routing, security and policy capabilities.

Most of these features are being addressed by separate standardization efforts, such as WS-Coordination (Cabrera, 2002), WS-Transaction Management (Bunting, 2003), WS-BPEL (Arkin, 2005), WS-Addressing (Box, 2004), WS-Security (Atkinson, 2002), and WS-Policy (Bajaj, 2004). As the technological landscape becomes cluttered with acronyms and different standards, it might seem that the original goal of dynamic B2B integration may take long to achieve. Truth is however that most, if not all, challenges have been already identified, and several authors have introduced advances to Web service technology in features such as negotiation, contracting, security, matchmaking, monitoring, composition and coordination, as we will show in this chapter.

There is not much more to wait on the way towards supporting dynamic B2B integration – what we call business networking – if only one is able to sort out and combine the available contributions in Web service technology. This is precisely what we aim at in this chapter, by describing the contributions that are most relevant to supporting the full life cycle of business collaborations. Rather than reiterating how Web Services can support B2B exchanges – solutions for that have been available for a long time – our purpose is to introduce a full life-cycle approach, showing how Web Services can support every phase of business collaborations, from the time business partners search for each other to the time they perform and evaluate their interactions.

THE CONCEPT OF LIFE CYCLE IN INTEGRATION

Many solutions for B2B integration focus on defining the protocols and specifying the message format between business partners. It is clear that these frameworks concern the operation phase when two or more organizations, that have previously met, interact with each other. But business networking concerns more than just the operation phase: there must be mechanisms to search and find business partners, to establish contracts with them, to monitor exchanges, and to evaluate each other's performance. Operation is therefore just a single phase within a broader life cycle that includes search, selection, contracting, operation, and evaluation (Ferreira, 2004).

The concept of having a life cycle which requires different integration mechanisms across different life-cycle phases is known since the beginning of enterprise integration as a discipline. The reference architecture CIMOSA (AMICE, 1993), one of the first major enterprise integration architectures, established the distinction between a "system life cycle" and a "product life cycle", the latter unfolding within the operation phase of the former. Then GERAM (IFIP-IFAC, 1999), the generalized reference architecture, generalized the concept of nesting the life cycle of different entities within the operation phase of one another.

The same approach can be applied to develop a framework for business networking. B2B exchanges take place within the operation phase of a larger life cycle comprising also search, selection, contracting, and evaluation. We will call this the "B2B trading life cycle" (Ferreira, 2004). In this larger life cycle, B2B exchanges are no longer fixed by any particular B2B standard. Rather, the conversations that take place during the operation phase and between different organizations depend upon the interactions that those organizations have agreed to perform. Defining these interactions is part of the B2B trading life cycle, and it happens during the contracting phase.

But then, depending on the markets that organizations find themselves in, the B2B trading life cycle may include these or other phases. And because a company may outsource some of its activities or associate itself with other partners according to an arbitrary structure, the selection and contracting phases for this company may be more elaborate than for other participants. So how can an organization learn about the B2B trading life cycle that takes place in a particular market? It becomes apparent that the B2B trading life cycle, on its turn, is just a single phase within an even broader life cycle – the business networking life cycle – which describes how organizations can enter a market and obtain information about how business networks are developed within that market. Figure 1 illustrates the relationships between these different life cycles.

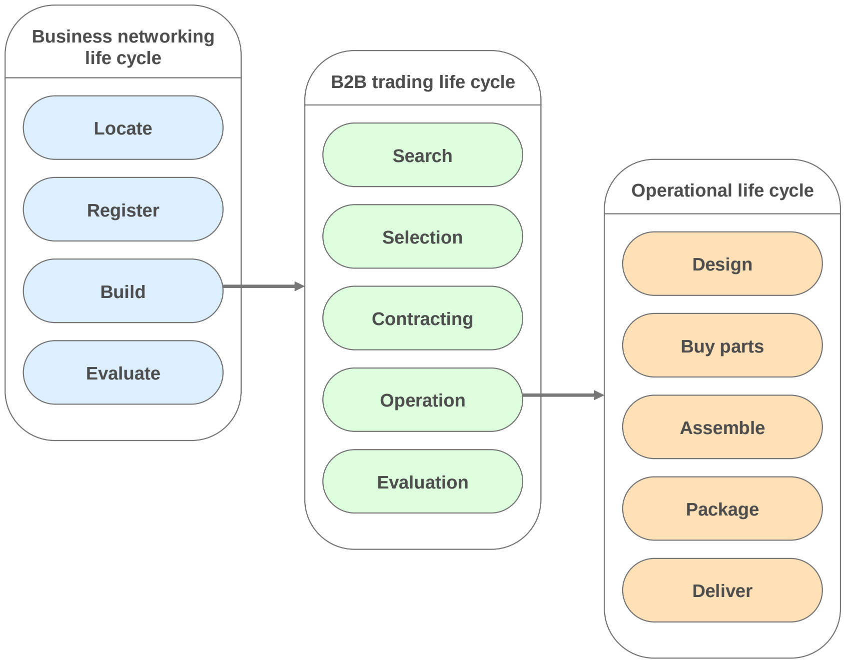

Figure 1. The business networking life cycle hierarchy

In the following sections, we begin by describing how Web Services support the operational life cycle. Then, we will refer to a series of recent advances that can be brought together to support the typical phases of a B2B trading life cycle. Finally, we explain how Web Services can be used to support the business networking life cycle as well. We will conclude that all these life cycles can be supported by recursively applying the basic principles of Web Services technology.

SUPPORTING THE OPERATIONAL LIFE CYCLE

Supporting B2B operational exchanges has always required companies to incur in costly system integrations. Even the adoption of XML and B2B frameworks (such as RosettaNet) has not dramatically reduced the amount of effort required to attain B2B exchanges, since it is still necessary to integrate the B2B side with internal backoffice systems. This is often done by interfacing B2B systems with the enterprise messaging bus, which requires customized interfaces between different systems and produces tightly-coupled solutions. These solutions will have to be reworked as soon as the partners, the conversation protocols or the supporting systems change.

Service-Oriented Architectures (SOAs) are known to alleviate this problem by allowing more flexible and loosely-coupled architectures. An appropriate way to implement a service-oriented architecture is precisely by taking advantage of Web Services. The key element that makes Web Services so interesting for this purpose is the ability to search for and bind to services both at build-time and run-time. As shown in figure 2, besides the service requester and provider, there is a service registry that allows service discovery and interaction, without the need to have services bound to each other to start with.

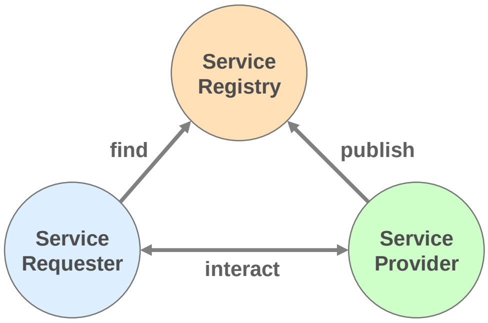

Figure 2. The SOA triangle (Champion, 2002)

This approach provides a level of indirection that was previously unavailable, since B2B conversations no longer have to be hard-wired. Instead, the service requestor can look up the conversation protocol in the service registry, which can be given as a Web Services Conversation Language (WSCL) description (Banerji, 2002). WSCL is able to specify a set of interactions (one-way or two-way), how these interactions follow one another, and what are the document formats (XML Schemas) used in each interaction. Configuring a B2B conversation is therefore no longer a programmer's task; rather, it can be readily supported by the underlying service infrastructure.

Once the conversation is taking place, an open issue is how to correlate requests and responses without mixing them with messages from other simultaneous conversations. Brambilla et al (2002) discuss five different correlation mechanisms and present the advantages and disadvantages of each of them. These mechanisms include transport-level correlation, which requires asynchronous transport protocols not widely supported, and application semantics correlation, which requires shared data structures. They also include correlation mechanisms are based on metadata, to be included in every message regardless of its particular format. The metadata relates each message to a particular operation type and instance, and possibly also to a particular conversation type and instance.

An alternative approach to specifying conversations as a set of one-way or two-way operations is, as in (Hanson, 2002), to describe them as a complete state machine together with the operations that trigger changes in state. This state machine is referred to as the *conversation policy*. The main advantage of this approach is to be able to nest an entire conversation policy into a single state of another one, allowing conversations to be described by a composition of sub-conversations. This may be useful in order to be able to reuse sub-conversations or change them without impacting the overall behavior.

SUPPORTING THE B2B TRADING LIFE CYCLE

In addition to the operation phase, in the B2B trading life cycle we are interested in supporting partner search and selection, contracting, configuration, and evaluation. Supporting these phases requires different kinds of mechanisms, but the underlying infrastructure can be the same as before:

- Partner search and selection involves querying service registries and applying search criteria in order to match goals, processes, or even QoS (Quality of Service) parameters.

- Contracting requires facilities to support negotiation, which can be regarded as a kind of B2B conversation, and languages to describe service-level agreements (SLAs).
- Configuration deals with making resources properly accessible (e.g. via Web Services) in order to carry out the agreed B2B conversations during the operation phase.
- Evaluation may involve active monitoring during the operation phase and its ultimate purpose is to assess the performance of the operation phase.

There have been recent advances in Web Services technology that are relevant to all of these phases. We will organize the discussion by dividing these contributions according to the B2B trading phase where they seem to be most useful.

Supporting Partner Search and Selection

One if the first tasks in the B2B trading life cycle is to look for potential business partners that will satisfy given criteria. Since Web Services technology inherently provides publishing and discovery capabilities, it is convenient to use those capabilities (notably UDDI) to search for business partners. However, UDDI registries are centralized repositories. Unless there is a global agreement on which repository to use, there will always be multiple UDDI registries that could be potentially relevant to the partner search being performed.

Zhang et al (2003) address the problem of searching in multiple UDDI registries and aggregating the search results. They do this by expressing search queries in a UDDI Search Markup Language (USML), which can be used to specify different registries, the queries to be performed in each registry, and the aggregation operator that determines how the search results are to be consolidated (this can be as simple as an AND or OR operator). The USML is passed on to a search engine which parses the query, builds search commands, dispatches them to UDDI registries and finally combines the responses and sends the end result back to the requester.

Whereas USML makes use of UDDI data elements to specify queries, other authors have proposed the use of more sophisticated criteria. Dumas et al (2004) propose a framework to describe trading intentions as constraints on attributes of a given schema (ontology). The example given is that of a car buyer who wants to find a car of a certain model, not older than a number of years and within a certain price range, where model, year and price are attributes defined by the given schema. The matchmaking with the trading intention of a car seller (or anyone else, for that matter) is done by satisfying the conjunction of the constraints in the two intentions.

Besides satisfying goals, a company may actually want a business partner that carries out their activities in a particular way. In this case, the challenge is to be able to find business partners that comply not only with a given trading intention but also with a given message exchange sequence. Wombacher et al (2003) introduced such an approach, where finite state automata (FSA) are used to describe message-exchange sequences. Basically, the proposed algorithm computes the intersection of two FSA, which describe both the desired sequence and the provided one. It is shown that if the intersection contains an empty automaton then the two sequences do not match, otherwise they are compatible. The use of FSA is motivated by the fact that they can be mapped to WSCL in a straightforward way.

Other search criteria may require potential business partners to comply with given QoS preferences. As with trading intentions, matching QoS preferences can be done with constraint satisfaction algorithms. For example, Lin et al (2005) employ fuzzy logic to express QoS preferences and a branch-and-bound algorithm to support service composition. This composition is achieved by selecting candidate Web Services with different QoS criteria values. Zeng et al (2004) propose a set of concrete QoS criteria to be used in service composition: execution price, execution duration, reputation, successful execution rate, and availability. They also discuss how to implement QoS-driven service selection either by local optimization (selecting the one from a set of similar services) or by global planning (selecting the services that optimize the overall composition).

In any case, it is necessary to make sure that the selected services are actually provided the required input in order to carry out their job, and that they are actually able to produce the desired outputs. In addition, there may be a set of preconditions that must be satisfied so that the service can be performed. Tomaz et al (2003) propose a system that can deal with inputs, outputs and preconditions at the semantic level, using the DAML-S (now OWL-S) ontology language to express buyer requests and supplier advertisements. The matchmaking is done by a semantic engine which, based on the given ontology, will determine the best fit by computing the distance between different concepts in the taxonomy tree.

Supporting Negotiation and Contracting

Despite the fact that business contracts must be objective, unambiguous and complete, the negotiation that precedes a contract is to a large extent an unstructured process, in the sense that it may require an unspecified number of interactions until all parties are satisfied with the result. This poses a challenge to the automation of negotiation and contracting. It is apparent, however, that most negotiation processes end up following similar patterns (Robinson, 1998) and that these patterns can be supported by Web Services technology in different ways.

Chiu et al (2005) propose a methodology to manage negotiation processes, which has been implemented with Web Services. The methodology relies on contract templates, which can either refer to the negotiation process of a previously established contract or to a new negotiation process to be supported. Once the contract template has been created, its execution is supported by a Negotiation Support System (NSS), which provides the Web Services required for each activity in the negotiation process. For example, in some negotiations dealing with requests for proposals there is a Web Service for placing the request and another Web Service for suppliers to submit the proposals. Negotiation processes based on bargaining and auctions are supported as well.

The same degree of automation may be achieved as suggested by Kim et al (2003) who explore the possibility (and problems) of describing negotiation processes with BPEL (Business Process Execution Language for Web Services). Even though the authors focus only the abstract process and not on creating an executable process model, it is evident that BPEL can be used to support the automation of a negotiation process by the orderly invocation of Web Services residing at different organizations. The approach is illustrated in an RFQ (request for quote) scenario.

Some authors (Lamparter, 2005) actually argue that full automation of contracting is not possible, where full automation is understood as contracting without any human intervention. It should be noted, however, that it is possible to automate with human intervention: for example, the invocation of a Web Service may result in placing a new task in the to-do list of an employee; when the task is done the same or another Web Service may send a reply back to the requester. The point of Lamparter et al (2005) is that contracting may involve negotiable as well as non-negotiable clauses, which can only be evaluated by humans. Therefore, they propose an ontology-based modeling framework that is directed at negotiable clauses only.

As in partner search and selection, contracting may also involve the specification of QoS parameters. In this context, the contracts are usually referred to as Service-Level Agreements (SLAs). Sahai et al (2002) discuss how to describe an SLA as a set of Service-Level Objectives (SLOs). An SLO is a specification of what parameter is to be measured, when it should be measured, how it is to be measured and evaluated, and what action should be done after measuring. One of the most common parameters is the client-side response time (Debusmann, 2004). This is usually measured by "instrumenting" the Web Service or the Web Service platform, as we will see ahead when discussing support for monitoring.

The negotiation of QoS parameters can be automated if QoS information about service providers is readily available. This information may be published in the UDDI registry in the form of WSOL documents (Tosic, 2003), an extension to WSDL that is able to express QoS constraints and combine them into boolean expressions. Besides constraints on QoS attributes, it will be useful to include the parameters of utility functions (Comuzzi, 2005) that specify how much a value of a QoS parameter is worth to the service requester or provider. If both the QoS information from the requester and from the provider are available, the negotiation can be fully automated by a third-party broker. If only the QoS information from the provider is available, it is still possible to make use of semi-automated negotiation (Comuzzi, 2005), which requires interacting with the service requester.

Supporting Configuration and Pre-Operation

From a technological point of view, a contract can be especially useful if it specifies the B2B conversation to take place between business partners during the operation phase. If such information can be gathered during the contracting phase then it is straightforward to express it using WSCL, BPEL or another orchestration language. In order to carry out that B2B conversation, however, partners must have their service platforms properly configured. This particular task can also be automated, even if partners have different infrastructures: the task of generating Web Service wrappers from the specification of a B2B conversation is similar to that of generating Web Service proxies/stubs from a WSDL description. In scenarios where the Web Service infrastructure is already in place, configuration may turn into a problem of service composition. Several authors have explored these different possibilities.

Baghdadi (2004) addresses the problem of generating Web Services from coordination requirements. The approach is holistic in the sense that it starts by creating an overall business model of the main systems in an enterprise information infrastructure. This business model is expressed in terms of business objects, events, processes and states – the so-called "elements of discourse". Then the concept of "factual dependency" is introduced: what elements must be created/updated/removed/retrieved when a given event occurs. The proposed approach is based on generating a Web Service that implements each factual dependency. The same approach could be used to generate the Web Services that implement a B2B conversation, described as a set of factual dependencies between different business partners.

Shan et al (2005) get closer to achieving this goal using workflow views rather than factual dependencies. A workflow view is a flow graph of activities with input and output messages, and it is a publicly available subset of the actual process taking place within an enterprise. A B2B conversation is the result of connecting workflow views of different partners via message exchange, and it is described with BPEL. These conversations are then orchestrated by a workflow-based B2B process engine. It is interesting to note that the proposed system supports the generation of Web Service definitions (in WSDL) from workflow views, where each activity is mapped onto a WSDL port. This effectively provides a jump start to the configuration of B2B interactions.

Other authors address precisely the inverse problem: if the Web Service infrastructure is already in place, and there are multiple service offerings to choose from, then what B2B interaction should be used to achieve a given business goal? This problem falls within the scope of dynamic service composition (Tosic, 2001) and an effective way to solve it is to rely on planning techniques (Madhusudan, 2004). According to this approach, each Web Service is modeled as a collection of elementary operations as specified in the corresponding WSDL description. The point is that some service requests may be satisfied by invoking an elementary operation, while others require the invocation of several operations from the same or different Web Services. In the latter case, the sequence of required operations and Web Services is computed via Hierarchical Task Network (HTN) planning. The same technique could possibly be used to determine the B2B conversation that must be carried out in order to satisfy the business goals expressed in a contract.

Supporting Monitoring and Evaluation

The last phase of the B2B trading life cycle is the evaluation phase, when business partners assess the performance of each other and gather information that may be useful in future partner selections. Evaluation is related to decision-making, so the opportunity for technological support is in focusing on gathering information to facilitate evaluation. This explains why monitoring is so relevant for the purpose of evaluation. Thus, even though monitoring takes place during the operation phase, its purpose is to support the evaluation phase. Among other issues, recent advances in monitoring have addressed performance evaluation, compliance with SLAs, data quality management, and trustworthiness of the service infrastructure.

Monitoring can be done at the lowest level of the Web Services stack by intercepting and logging every SOAP message (Cruz, 2004), which may be useful to extract information about service use and performance. A more convenient way to obtain such information is by "instrumenting" the Web Services platform. Debusmann et al (2004) illustrate how to instrument the Axis platform using aspect-oriented programming. Since there are two separate request and response handlers, it is possible to measure the client-side response time by means of the interval between their invocation, for a given pair of correlated request and response. McGregor et al (2003) make use of instrumentation in order to maintain an audit-trail of all status changes during B2B conversations, described as WSFL flows (Leymann, 2001). Sahai et al (2002) make use of instrumentation in order to monitor SLAs and to record any violation of SLOs.

Monitoring and evaluation become an even more pressing need when the end product of a B2B collaboration must comply with quality management standards. Whereas the scope of these standards is traditionally confined to the environment of a single organization, B2B collaborations require quality management techniques to be extended to inter-organizational settings. This has already been approached for the case of information products (Shankaranarayanan, 2005) by means of a data quality metadata specification. The metadata associated with an information product describe how data flows and is processed across business partners to create that information product. The metadata is shared among business partners by means of a Web Service architecture which collects and provides metadata upon request. This architecture makes use of a central metadata exchange (MX) service which retrieves and consolidates metadata provided by the MX services running at each organization.

Another important role of the evaluation phase is to contribute to the development of trust among business partners. Since this subject goes far beyond the scope of Web Services technology, we will refer only to the problem of improving the trustworthiness of the underlying service infrastructure. Zhang et al (2004) propose a framework to control the trustworthiness of computing in Web Services. Basically, the framework requires certain pieces of information to be included in the WSDL description of a Web Service. This information concerns resources (roles and how to invoke them), policies (e.g. security policies), validation (ensuring predictability and handling exceptions), and management (tracking and monitoring). Resource, policy and validation info are expressed using WS-Resource, WS-Policy and BPEL, respectively, while trustworthy management depends on the monitoring capabilities of the run-time platform.

SUPPORTING THE BUSINESS NETWORKING LIFE CYCLE

Much like the eCo framework once proposed by Tenenbaum et al (1997), which comprised several layers ranging from services up to markets and networks (eCo Working Group, 1999), the business networking life cycle can be regarded as the top layer above the B2B trading life cycle. In the B2B trading life cycle discussed previously, organizations search for and select business partners, and settle, configure and evaluate the interactions that take place with those partners. These tasks are just a single phase within the business networking life cycle, as shown in figure 1, where companies locate and enter markets in order to build trading relationships with other players in that market.

In the previous section we discussed the latest developments towards providing Web Service infrastructures with search, contracting and monitoring capabilities, among others. Now, regarding the business networking life cycle, the required capabilities are those that provide companies with the ability to organize themselves into markets and to join and develop collaborations within those markets. A market may be either vertically-oriented, being industry- or product-specific, or horizontally-oriented, dealing with goods and services that are common to multiple industries (Sahai, 2001). In addition, each market may have its own set of rules, such as how to advertise purchase needs, how to select partners, how perform negotiations, etc.

The way a company enters a market and starts interacting with other players in that market resembles the way a peer enters a P2P network, joins a peer group and starts interacting with the peers within that group. In the JXTA platform (Traversat, 2002), for example, peers join peer groups where special-purpose services are available. Peer groups are described by XML advertisements published in the network, and services are described by advertisements published within a peer group. Both kinds of advertisements can be found by means of a discovery service, and they contain the necessary information to bind and interact with other resources (peers and peer groups).

The analogy between peer groups and electronic markets is striking, as well as between JXTA service advertisements and WSDL definitions, and the JXTA discovery service can be seen as a decentralized version of the service registry concept. Therefore, Web Services technology does provide the required features for a business networking infrastructure, although not as decentralized as in a P2P platform, since Web Services rely on a centralized service registry. In the business networking life cycle, the role of the service registry is to allow companies to find markets and, once they have entered a market, to retrieve information about the trading life cycle that applies within a market – whether it is the five-phase life cycle described earlier or any other kind of service choreography. This choreography can be described by means of a WSCL or WSFL, for example, and it can be retrieved from the service registry within the market.

Albrecht et al (2004) make a comparison of five different approaches towards supporting electronic marketplaces, and Web Services stand out as an appropriate infrastructure to support loosely connected marketplaces, being hampered only by the lack of standards. As standardization is evolving in several fronts, a more severe handicap may be the fact that Web Services technology relies on centralized services, so it is still not entirely clear how to make use of it in highly distributed environments (Huang, 2003). Nevertheless, the developments described in the previous sections span across all the different phases of business collaborations, from the business networking life cycle down to the operational life cycle. Figure 3 illustrates the role that features of Web Services technology can play each of these phases.

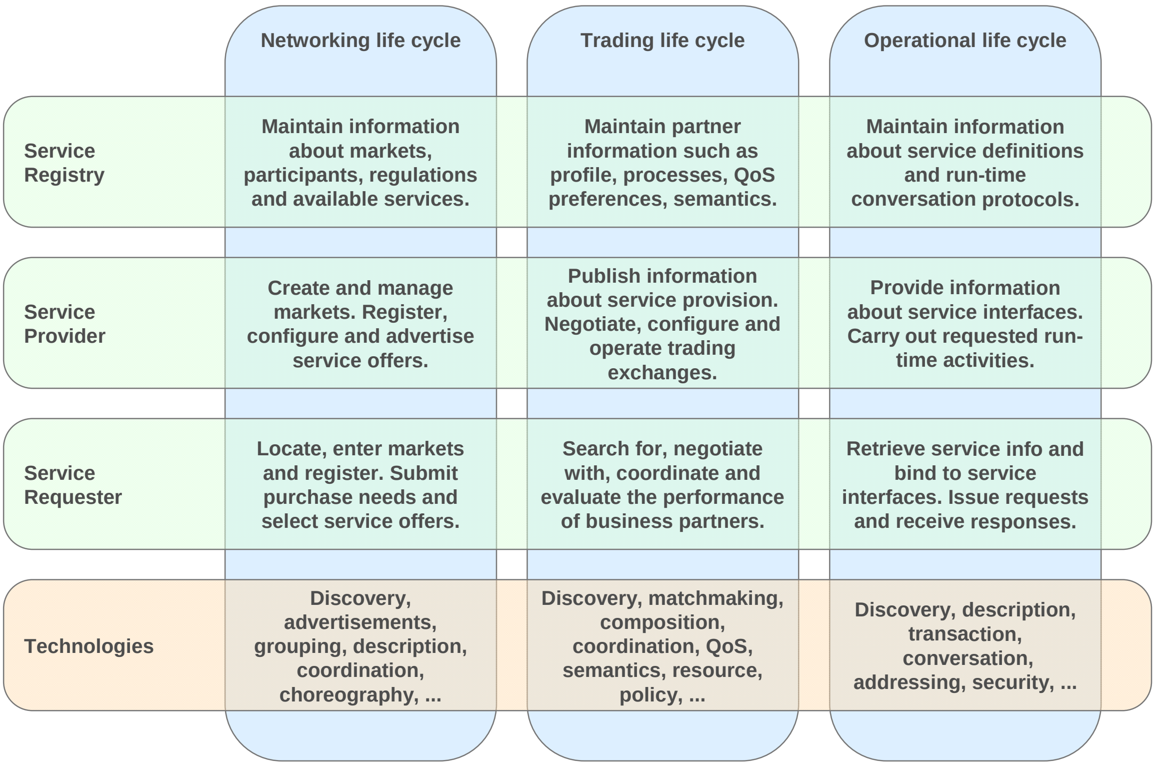

Figure 3. The role of Web Services technology in B2B collaborations

As suggested in figure 3, a large amount of information stored in the service registries concerns the description of B2B conversations to take place between business partners. This means that Web Services technology, besides providing dynamic binding via a service registry, should also include process management capabilities in order to support the design, execution and monitoring of those conversations. In fact, a survey of Web Service architectures (Myerson, 2002) shows that key technology vendors intend to provide workflow capabilities on top of their Web Services stack. This may be eased by the fact that Web Services seem to be a suitable foundation for business process management both within and across enterprise boundaries (Leymann, 2002).

As an example of how such capabilities can be implemented on top of Web Services, Gomez et al (2005) propose a B2B conversational architecture organized in four layers. The Web Services Layer encompasses basic features such as publishing and searching for service information in a UDDI registry. The Composite Web Service Layer makes use of a second repository, which keeps information about available composite services, with compositions being described by means of BPEL documents. The Enterprise Intra-Workflow Layer deals business processes that are described as orchestrations of composite services, and stored in a third repository as WS-CDL documents (Kavantzas, 2004). The top-most layer is the Enterprise Inter-Workflow Layer which makes use of a fourth repository in order to store rules, policies and regulations that govern B2B interactions. Except for the bottom Web Service Layer, each layer is associated with an engine that provides execution and monitoring capabilities.

CONCLUSION

In the operational life cycle, Web Services support B2B exchanges as they encapsulate the business activities within a B2B collaboration. In the B2B trading life cycle, Web Services also encapsulate business activities, but now the purpose of these activities is to search for and select business partners, and to settle, configure and evaluate the interactions that take place with those partners. In the business networking life cycle, Web Services will again encapsulate a set of business activities. These business networking activities deal with locating and entering markets, where a company will build relationships and undertake collaborations with other players in that market.

Web Services technology is pervasive in the way that it is able to provide a common infrastructure that supports B2B conversations at each of these different levels. The key enabling features are the same across these levels: the ability to publish, search for and interact with remote services. This chapter has drawn attention to the latest developments in Web Services technology that are paving the way towards supporting the full life cycle of business collaborations. Clearly, there is a wide range of research challenges that still lack a definite answer. Yet, Web Services have already succeeded in gathering a community that is developing the capabilities required to turn Web Services into a business networking platform.

REFERENCES

Albrecht, C., Dean, D., & Hansen, J. (2005). Marketplace and technology standards for B2B e-commerce: progress, challenges, and the state of the art. *Information & Management*, 42(6).

AMICE ESPRIT Consortium (1993). *CIMOSA: Open System Architecture for CIM*. Springer-Verlag.

Arkin, A., Askary, S., Bloch, B., Curbera, F., Goland, Y., Kartha, N., Liu, C., Thatte, S., Yendluri, P., & Yiu, A. (2005). *Web Services Business Process Execution Language*, OASIS Committee Draft.

Atkinson, B., Della-Libera, G., Hada, S., Hondo, M., Hallam-Baker, P., Kaler, C. (ed.), Klein, J., LaMacchia, B., Leach, P., Manferdelli, J., Maruyama, H., Nadalin, A., Nagaratnam, N., Prafullchandra, H., Shewchuk, J., & Simon, D. (2002). *Web Services Security (WS-Security)*. IBM, Microsoft, and Verisign.

Baghdadi, Y. (2004). A business model for B2B integration through Web services. In *Proceedings of the IEEE International Conference on e-Commerce Technology* (pp.187-194).

Banerji, A., Bartolini, C., Beringer, D., Chopella, V., Govindarajan, K., Karp, A., Kuno, H., Lemon, M., Pogossiants, G., Sharma, S., & Williams, S. (2002). *Web Services Conversation Language (WSCL) 1.0*. W3C Note.

Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Malhotra, A., Nadalin, A., Nagaratnam, N., Nottingham, M., Prafullchandra, H., von Riegen, C., Schlimmer, J. (ed.), Sharp, C., & Shewchuk, J. (2004). *Web Services Policy Framework (WSPolicy)*. BEA, IBM, Microsoft, SAP, Sonic, Verisign.

Box, D. (ed.), Christensen, E., Curbera, F. (ed.), Ferguson, D., Frey, J., Hadley, M., Kaler, C., Langworthy, D., Leymann, F., Lovering, B., Lucco, S., Millet, S., Mukhi, N., Nottingham, M., Orchard, D., Shewchuk, J., Sindambiwe, E., Storey, T., Weerawarana, S., & Winkler, S. (2004). *Web Services Addressing (WS-Addressing)*. BEA, IBM, Microsoft, SAP, Sun.

Brambilla, M., Ceri, S., Passamani, M., & Riccio, A. (2004). Managing asynchronous Web services interactions. In *Proceedings of the IEEE International Conference on Web Services* (pp.80-87).

Bunting, D., Chapman, M., Hurley, O., Little, M. (ed.), Mischkinsky, J., Newcomer, E. (ed.), Webber, J., & Swenson, K. (2003). *Web Services Transaction Management (WS-TXM)*. Arjuna, Fujitsu, IONA, Oracle, Sun.

Cabrera, L. F., Copeland, G., Feingold, M. (ed.), Freund, R., Freund, T., Johnson, J., Joyce, S., Kaler, C., Klein, J., Langworthy, D., Little, M., Nadalin, A., Newcomer, E., Orchard, D., Robinson, I., Shewchuk, J., & Storey, T. (2005). *Web Services Coordination (WS-Coordination)*. Arjuna, BEA, Hitachi, IBM, IONA, Microsoft.

Champion, M., Ferris, C., Newcomer, E., & Orchard, D. (2002). *Web Services Architecture*. W3C Working Draft.

Chinnici, R., Moreau, J.-J., Ryman, C., & Weerawarana, S. (2005). *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. W3C Working Draft.

Chiu, D., Cheung, S., Hung, P., Chiu, S., & Chung, A. (2005). Developing e-Negotiation support with a meta-modeling approach in a Web services environment. *Decision Support Systems*, 40(1), 51-69.

Clement, L., Hately, A., von Riegen, C., & Rogers, T. (eds.) (2004), *UDDI Version 3.0.2*, OASIS UDDI Spec Technical Committee Draft.

Comuzzi, M., & Pernici, B. (2005). An Architecture for Flexible Web Service QoS Negotiation. In *Proceedings of the 9th IEEE International EDOC Conference*.

Cruz, S. da, Campos, M., Pires, P., & Campos, L. (2004). Monitoring e-business Web services usage through a log based architecture. In *Proceedings of the IEEE International Conference on Web Services* (pp.61-69).

Debusmann, M., & Geihs, K. (2004). Towards dependable Web services. In *Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing* (pp.5-14).

Dumas, M., Benatallah, B., Russell, N., & Spork, M. (2004). A configurable matchmaking framework for electronic marketplaces. *Electronic Commerce Research and Applications*, 3(1), 95-106.

eCo Working Group (1999). *eCo Architecture for Electronic Commerce Interoperability*. CommerceNet.

Ferreira, D., & Ferreira, J. J. (2004). Building an e-marketplace on a peer-to-peer infrastructure, *International Journal of Computer Integrated Manufacturing*, 17(3).

Gomez, J., Hernandez, G., Olmedo, J., & Bussler, C. (2005). A B2B Conversational Architecture for Semantic Web Services Based on BPIMS-WS. In *Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems* (pp.252-259).

Hanson, J., Nandi, P., & Levine, D. (2002). Conversation-enabled Web Services for Agents and e-Business, In *Procedings of the International Conference on Internet Computing*.

Huang, Y., & Chung, J.-Y. (2003), A Web services-based framework for business integration solutions. *Electronic Commerce Research and Applications*, 2(1), 15-26.

IFIP-IFAC Task Force (1999). *GERAM: Generalised Enterprise Reference Architecture and Methodology*. Version 1.6.3.

Kavantzas, N., Burdett, D. & Ritzinger, G. (2004). *Web Services Choreography Description Language Version 1.0*. W3C Working Draft.

Kim, J., Segev, A., Patankar, A., & Cho, M. (2003). Web Services and BPEL4WS for Dynamic eBusiness Negotiation Processes, In *Proceedings of the International Conference on Web Services*.

Lamparter, S., Mutschler, S., Luckner, S., Stockmar, K., & Laborde, C. (2005). A Modeling Perspective on Web Service Contracting. In *Proceedings of the Workshop on Contract Architectures and Languages*, 9th IEEE International EDOC Conference.

Leymann, F. (2001). *Web Services Flow Language (WSFL) Version 1.0*. IBM Software Group.

Leymann, F., Roller, D., & Schmidt, M.-T. (2002). Web services and business process management, *IBM Systems Journal*, 41(2).

Lin, M., Xie, J., Guo, H., & Wang, H. (2005). Solving QoS-driven Web service dynamic composition as fuzzy constraint satisfaction. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service* (pp.9-14).

Madhusudan, T., & Uttamsingh, N. (2004). A declarative approach to composing web services in dynamic environments, *Decision Support Systems* (to appear).

McGregor, C., & Scheifer, J. (2003), A framework for analyzing and measuring business performance with Web services, In *Proceedings of the IEEE International Conference on E-Commerce* (pp.405-412).

Myerson, J. (2002). *Web Service Architectures*. Tect.

Österle, H., Fleisch, E., & Alt, R. (2001). *Business Networking: Shaping Collaboration Between Enterprises*, Springer.

Robinson, W. & Volkov V. (1998). Supporting the negotiation life cycle. *Communications of the ACM*, 41(5), 95-102.

Sahai, A. & Machiraju, V. (2001). Enabling of the Ubiquitous e-Service Vision on the Internet. *e-Service Journal*, 1(1).

Sahai, A., Machiraju, V., Sayal, M., Jin, L., & Casati, F. (2002). *Automated SLA Monitoring for Web Services*. HP Technical Report HPL-2002-191.

Sakata, Y., Yokoyama, K., & Matsuda, S. (2004). A method for composing process of nondeterministic Web services. In *Proceedings of the IEEE International Conference on Web Services* (pp.436-443).

Shan, Z., Chiu, D., & Qing, L. (2005), Systematic Interaction Management in a Workflow View Based Business-to-Business Process Engine. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*.

Shankaranarayanan, G., & Cai, Y. (2005), A Web Services Application for the Data Quality Management in the B2B Networked Environment, In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*.

Shim, S., Pendyala, V., Sundaram, M., & Gao, J. (2000). Business-to-Business E-Commerce Frameworks, *IEEE Computer*, 33(10).

Tenenbaum, J., Chowdhry, T., & Hughes, K. (1997). Eco System: An Internet Commerce Architecture. *IEEE Computer*, 30(5), 48-55.

Tomaz, R., Labidi, S., & Wanghon, B. (2003). A semantic matching method for clustering traders in B2B systems. In *Proceedings of the First Latin American Web Congress* (pp.144-153).

Tosic, V., Pagurek, B., Esfandiari, B., & Patel, K. (2001). On the Management of Compositions of Web Services. In *Proceedings of the Object-Oriented Web Services Workshop*, ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications.

Tosic, V., Pagurek, B., Patel, K., Esfandiari, B. & Ma, W. (2003). Management Applications of the Web Service Offerings Language (WSOL), *Lecture Notes in Computer Science*, 2681, 468-484.

Traversat, B., Abdelaziz, M., Duigou, M., Hugly, J.-C., Pouyoul, E. & Yeager, B. (2002). *Project JXTA Virtual Network*. Sun Microsystems.

Vinoski, S. (1997). CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments. *IEEE Communications Magazine*, 14(2)

Wombacher, A., Fankhauser, P., Mahleko, B., & Neuhold, E. (2003). Matchmaking for business processes, In *Proceedings of the IEEE International Conference on E-Commerce* (pp.7-11).

Zeng, L., Benatallah, B., Ngu, A., Dumas, M, Kalagnanam, J., & Chang, H. (2004). QoS-aware middleware for Web services composition. *IEEE Transactions on Software Engineering*, 30(5), 311-327.

Zhang, L., Chao, T., Chang, H., & Chung, J.-Y. (2003). XML-Based Advanced UDDI Search Mechanism for B2B Integration, *Electronic Commerce Research*, 3(1-2), 25-42.

Zhang, J., Zhang, L.-J., & Chung, J.-Y. (2004). WS-trustworthy: a framework for Web services centered trustworthy computing. In *Proceedings of the IEEE International Conference on Services Computing* (pp.186-193).