ABBA: an architecture for deploying business-to-business electronic commerce applications

Yousef Baghdadi *

Department of Mathematics and Computer Science, UAE University, P.O. Box 17551, Al-Ain, UAE

Received 5 October 2002; received in revised form 15 September 2003; accepted 19 September 2003

Abstract

An efficient design process for developing B2B EC applications is critical for numerous reasons, namely: (i) the complexity and the growth of this category of EC, (ii) its differences from other categories in many aspects, and (iii) the number of existing approaches and standards causing confusion to enterprises willing to deploy B2B EC applications. In this paper, we first analyze and compare existing architectures, reference models, approaches, standards and languages used to deploy B2B EC applications. In the light of this analysis and a specification of B2B EC requirements, we propose ABBA, a comprehensive architecture, and a corresponding design process. ABBA is a layered architecture with four interfaced abstraction levels: business model and business process, business process decomposition and distribution, supporting services, and integration technology. ABBA aims at making a common view of the business process independent from the implementing technology, and providing guidelines for a design process to develop comprehensive, interoperable and scalable B2B EC applications.

1. Motivation

Electronic commerce (EC) is the ability to conduct business transactions via respective services and protocols (e.g., WWW, EDI, XML, HTTP and SOAP) of the Internet and VAN (Value Added Networks). EC is generally broken down into categories depending on the roles of the actors (e.g., business, customer, administration and employee). The well-known categories are business-to-consumer (B2C), which is selling goods and services directly to consumers; and business-to-business (B2B) which is related to businesses selling goods and services to other businesses. Regardless of its category, the value of EC is clear: reaching a wider market, saving time, cutting costs, responding to customer queries quickly, obtaining needed product data, making informed purchase decisions, and developing and maintaining relationships [36]. However, this work focuses particularly on developing and deploying B2B EC applications according to the following motivations:
• The growth of EC is mainly in B2B. Zwass [38] notes that B2B volume surpasses that of B2C by a factor of 8 or 9.
• B2B differs from B2C in many aspects, namely: the partner’s relationships and coordination mechanisms, such as electronic networks [33], the type of transactions (e.g., load, degree of integration and innovation), the content (e.g., products presentation), and the personalization (e.g., type of customization) [22].
• B2B differs from B2C in the deployment of the computational infrastructure (applications and platforms). While B2C requires some Web servers, backend applications and databases (business side) and a Web browser (consumer side), B2B integration requires a common agreed-upon business process, EAI (Enterprise Application Integration) of each of the involved partners and middleware [23], in addition to the services of the Internet and VAN.
• B2B applications deployment is complex and costly [16]. Linthicum [23] notes that though the value of B2B is clear, how to make it happen in the real world is much less clear. Moreover, the number of EC architectures, models, approaches and standards give us an idea of the complexity of such a deployment.

Now, EC applications implement only a portion of the market transaction (e.g., order process, payment), on a case-by-case basis, and with a focus on the technology. The question is “Can we succeed in developing an entire integrated B2B applications?” We argue that it is possible provided that companies avoid being driven by the current technology. That is, companies must focus first on the specification and (re-) engineering of the business process that crosses the boundaries of the organization, i.e., roles, functions [7], and requirements. We need to understand the flows of information between businesses and the barriers to these flows, in order to get a common business process. Mastering the comprehensive business process assists in deciding a suitable technology (existing technology or new one to develop) that makes the business process more innovative, integrated and efficient. Indeed, current technologies might not be suitable [35]. They are generally more developer-oriented than business-oriented [23].

Moreover, B2B EC applications developed on a case-by-case basis, focus only on a part of the B2B transaction. B2B EC applications which do not implement important phases (e.g., pre-transaction, post-transaction) and partners’ relationships (e.g., interaction aspects, namely mechanisms of coordination) are not very profitable.

Therefore, a framework with multiple abstraction levels is required to specify the whole business process (including its distribution over involved businesses) independently of the implementing technology. Indeed, this independent specification helps in deciding the right B2B integration technology.

This work proposes ABBA, an Architecture for deploying Business-to-Business Applications, an architecture with four interfaced abstraction levels: (1) business model and business process, (2) business process decomposition and distribution, (3) supporting services, and (4) integrating technology; and a consequent six-step design process to develop interoperable and scalable B2B EC applications.

The objectives of ABBA are manifold:
• Independence business process/technology. A design process of a B2B EC application must begin with the requirements, the analysis, the specification, and the modeling of the whole business process. The business process (flows of information and processes) must be understood and agreed by all involved businesses. The implementing technology must be decided later. Although, the technology is an enabler, there may be various implementing technologies for the same business process. Technology options include for instance, leveraging companies’ technologies, reusing market technologies (e.g., Web services, OMG objects), or completely developing new appropriate technologies [18,30]. This requires a common representation (or modeling), and a managing responsibility of the business process regardless of the technology.
• Guidelines for a design process (towards a methodology) to deploy B2B EC applications. Due to the inherent complexity and the important deployment costs of B2B EC applications, a development on a case-by-case basis is inefficient. We need a comprehensive design process. Such a design process assisted by models, for-
malisms, languages, and tools aims at reducing the complexity and cutting the development costs. ABBA provides these guidelines. For instance, we can envision each level or interface of ABBA as a step towards the deployment and maintenance of B2B EC applications.

- Communication between people involved in the design process (e.g., decision-makers, technologists and actors). Models, formalisms, and languages used at different levels, high (business process level) as well as low (technology level), allow communication between people “speaking” different domain-related languages.

The rest of this paper is organized as follows: Section 2 presents a deep study and a comparison of current architectures, reference models, approaches, standards, and languages for B2B EC. Section 3 specifies the desired properties and the requirements of B2B EC. Section 4 elaborates the components of ABBA. Section 5 details ABBA corresponding design process. Section 6 concludes the paper and suggests future developments and issues.

2. Approaches for B2B EC applications

This section investigates, analyses and compares (with respect to our goals) successively current architectures, reference models, approaches, standards, and languages for B2B EC.

2.1. Architectures and reference models

Being aware of the possible failures of EC deployment, the community of EC emphasizes on architectures and reference models to develop EC applications. This work considers most of the related architectures and reference models.

2.1.1. The eCo System

The eCo System [34] addresses the interoperability of EC applications. It was conceived originally as a CORBA-based interoperability framework. The eCo system architecture was recast in 1997 on XML foundation [12]. A key element of the eCo system is the CBL (Common Business Library). CBL is an extensible public collection of BIDs (Business Interface Documents) and document templates. BIDs, posted on the Web, tell potential trading partners what online services and what documents to use when invoking these services. A company, provided with an eCo server, can use CBL to create XML version of its forms and documents by customizing the building blocks included in the CBL. The eCo server is used to parse incoming documents, and to invoke the appropriate services.

The eCo System is a layered architecture (Fig. 1). Each layer models key business processes and services to build Internet markets. The vertical markets layer contains services specific to particular Internet markets (e.g., insurance, health). The business processes and applications layer contains generic business services common to multiple Internet markets. The commerce services allow individuals and companies to authenticate themselves and their partners. The network services layer contains services that enhance the performance, reliability, and security of the net.

2.1.2. OMG reference model

OMG architecture for EC [27] is based on the eCo System framework. It deals with an object-oriented architectural framework for Internet commerce that promotes the interoperation and the reuse of applications and services. OMG aims to make these services CORBA-compliant objects for interoperability. The framework (Fig. 2) groups related functional requirements into facilities which are categorized into three levels: market infrastructure services (catalogues, brokerage, and agencies), commerce facilities (contract service...
management, desktop facilities, and management of intellectual property rights), and low level services (payment, semantic data facility, profile management, and selection/negotiation). These levels have a common facility: the object browser and navigator (available to all facilities) which introduces an extensible framework for the inspection, presentation, and execution of EC entities.

2.1.3. EBES/EWOS building blocks for EC

The project of building blocks for EC (Fig. 3) is based on the concept of building blocks which identify the key technical components for EC [9]. The identification of such components is two-levels hierarchical decomposition. Commerce activities are partitioned into high level commercial processes: marketing, contracting, logistics, settlement, and interface with administrations. These processes are decomposed into lower-level activities denoted sub-processes (e.g., consult catalogues, quotation). A building block is a functional view of a sub-process. The implementing technology of a building block is called a solution, which is to be integrated into appropriate applications.

2.1.4. The electronic market-reference model

The electronic market-reference model (EMR-M) [32] represents two dimensions (Fig. 4). A horizontal dimension contains three phases of a market transaction: information, agreement and settlement. Whereas, the vertical dimension is made up of four views. The four views can be grouped into two main blocks. The upper two views focus on the organizational aspects (with rather long-term impacts). The lower two views depict technological aspects (with mainly short-term impacts).

2.2. B2B standards and languages

Several standards and languages have been developed to support interchange of information in B2B applications. Some of these standards such as EDI (Electronic Data Interchange), XML (eXtended Markup Language), ebXML (e-business eXtended Markup Language), and BizTalk are considered here generic standards. They can be used for horizontal as well as vertical markets. Whereas standards such as RosettaNet, OTP (Open Trading Protocol), ICE (Information and Content Exchange), and OBI (Open Buying on the Internet) are vertical market-oriented.

![Fig. 2. Principal facilities of OMG EC reference model.](image)

![Fig. 3. EBES/EWOS building blocks architecture.](image)
2.3. Web services

Web services are of particular interest for their ability to incorporate third-party applications or legacy systems [26]. Indeed, Web services are self-contained and modular applications that can be described, located, and invoked across the Web. They perform functions that can be anything from simple request to complicated business processes [17]. Web services implement the SOA (Service-Oriented Architecture) which provides a standard programming model. This programming model allows software components to reside on the network. They are described with WSDL (Web Services Definition Language), published and located through a UDDI (Universal Description, Discovery and Integration), and invoked over the Web through a SOAP (Simple Object Access Protocol), a standard used by applications to exchange XML documents through HTTP. SOA architecture provides BPEL4WS (Business Process Execution Language for Web Services), a process modeling language for Web services composition and aggregation. BPEL4WS focuses on a static composition of predefined flow of the processes and the bindings between them.

2.4. Agent approach

Software agent technology is an attractive paradigm to support EC applications. Indeed, agents can simulate the human behavior [24], buy and sell [25], or play the role of mediators [14]. In general, agents are intended to act on behalf of business actors to reduce the effort required to complete EC transactions. For instance, agents are intended to negotiate over multiple factors, including not only prices, but also tasks combination and temporal factors [5,31]. Agents can also be used to support supply chain management [19]. Agent-oriented techniques are used to support the modeling of virtual enterprises and their information systems [11]. Moreover, the mobility of agents increases the potential of applications in EC area, since mobile agents can autonomously decide to move from one actor to another.

2.5. Workflow approach

Some authors attempt to extend or enhance the WfMC architecture [37] with agent technology to adapt it to EC. For instance, Georgakopoulos et al. [10] deal with the problems associated with the development of multi-enterprise processes that explicitly capture and manage the services provided by each enterprise in a virtual space. Other approaches consist of adopting software agent for enhancing workflow systems to implement EC applications [8]. Two approaches for exploiting agent technology in the context of workflow are investigated: agent-enhanced workflow which aims at adding value to workflow systems by adding an agent layer responsible for social facilities, and
agent-based workflow which fully rethinks the system in terms of agents.

2.6. Brokerage systems

Several authors emphasize the importance of brokers for electronic markets [1–3]. Brokerage is a concept that aims to overcome limitations of the Internet commerce. Brokers are information-based actors aiming to shift important part of the business away from business’s partners. They are expected to provide some services to support a part of a transaction (e.g., search for products and partners, negotiate and agree on the contract, payment).

2.7. Integration approach

Some authors note that B2B relationships require a new type of integration technology that is dependent on flexible middleware that glue disparate applications, databases, and processes together [23]. For instance, b-cart (buyer cart) approach [13] is a an integration framework. Using the b-cart, a buyer can visit multiple e-marketplaces collecting items in his/her b-cart and make purchase order simultaneously over the collected items. By supporting integrated services (e.g., comparison on purchasing items, order tracking, payment), b-cart can be integrated with e-procurement systems including ERP (Enterprise Resources Management) systems in B2B EC.

2.8. Summary and comparison

2.8.1. Architectures and reference models

The choice of the above architectures and reference models is based on their perception, representation, modeling and composition of the business process. For instance, EBES/EWOS and EM-RM are rather business process-oriented. Accordingly, they present an implicit top-down approach starting from scratch to deploy EC applications. Therefore the problem of interoperability is not addressed as all their components are built in the same process. EBES/EWOS and EM-RM do not explicit how to define the building blocks and the business scenarios. OMG, eCo System are rather EC services-oriented. They concentrate on the interoperability between EC applications and services, and the reuse and composition of existing components. That is, the composition of EC services to build an application is done through an implicit bottom-up approach.

The studied architectures and reference models do not make explicit a design process to reduce the complexity of B2B EC applications where the concept of a common and comprehensive business process is critical.

2.8.2. Standards and languages

The variety of standards and languages cause confusion to companies facing a deployment of B2B applications because they do not make an explicit design process, namely for the horizontal markets. Indeed, the standards are mostly specific to vertical markets, where the business process is simple and understood by the involved partners (e.g., partner interface process of RosettaNet which is based on OBI). However, we note the relevance of XML since almost all the standards are based on XML. For instance, some generic standards based on XML such as Web services and ebXML provide models and languages for representing business processes. They accordingly tend to explicit a design process: top-down approach for ebXML and bottom-up approach for Web services (BPEL4WS).

2.8.3. Other approaches

Although workflow systems enable the automation of the business processes, they require end-user interactions at the run-times. They are still document-oriented exchange between end-users (not software components).

Agent technology is attractive to support EC applications as agents can model the behavior of human being. They are also used to enhance or extend workflow systems. However, agents paradigm cannot be accepted without a comprehensive

---

1 An attempt to analyze, evaluate and compare some reference models from the viewpoint of their suitability to support flexible and interoperable EC applications can be found in [28,29].
security framework, especially in the case of EC applications [6].

Brokers cause confusion as there are various types of brokers: international trade brokers, stock-brokers, real estate brokers, insurance brokers, advertising space brokers. Moreover, buyers can find much of the information on the Internet, they can bypass brokers and deal directly with the sellers.

Table 1 presents a comprehensive comparison with respect to criteria related to the objectives of ABBA, namely, the independence of business process specification from the implementing technology, and the design process. These criteria concern with: (1) the existence of a design process for B2B EC applications, (2) the business model dependence and specification. That is, whether or not the approach is specific to a business model (e.g., e-procurement, e-caution, third-party and so on), (3) the business process specification and modeling (independently from technology), (4) the market type dependence (e.g., vertical or horizontal), (5) the orientation (process-oriented or document-oriented), (6) the use of XML (due to the relevance of this standard), and (7) the requirements for the approach (e.g., eCo System requires an eCo server, OMG requires compliant-objects).

To conclude the comparison, we note that these approaches complete each other with respect to the abstraction levels and to the representation (or modeling) of the business process. For instance, EBES/EWOS and EM-RM present an abstraction level for the business process without considering the interoperability problem, which is solved by OMG. However, we still lack a coherent framework that guides almost all the issues of B2B EC. Therefore, a comprehensive architecture and a consequent design process are needed to develop scalable and interoperable B2B EC applications. This requires a comprehensive specification of the desired properties and the requirements of B2B.

3. Desired properties and requirements of B2B EC

The concepts of business process, business model, B2B process, coordination artifacts, B2B applications and systems, and their requirements are of paramount importance for ABBA architecture. They are specified in this section.

3.1. Definitions

3.1.1. Business process

A business process is an element of the value chain. Indeed, for organizational considerations, and with respect to a business model, the value chain is generally subdivided into primary business processes that participate directly to the creation of products or services (e.g., purchasing, manufacturing, fulfillment, and delivery), and supporting processes that assist the primary processes (e.g., recruiting employees, research and development) [15].

A business process is characterized by:
- Goals: a business process is intended to achieve defined objectives to create a value.
- Well-identified products and customers: the business objectives are matched through the business process and delivered in the form of products to the customers.
- Activities performed by different partners that collectively achieve defined goals of the business process.
- Derivation from a business model: a business process is always specified according to a business model.

3.1.2. Business model

A business model is the method used to conduct the business, i.e., selling and buying. Timmers [35] defines a business model as “an architecture for the products/services and information flows, including a description of the various business actors and their roles, a description of the potential benefits for the various actors, and a description of the sources of the revenues”. Accordingly, many types of business models (e.g., e-procurement, e-auction, brokerage model, virtual communities) are used in the market. The knowledge of the business model is critical to understand and derive the corresponding business processes and their requirements.
<table>
<thead>
<tr>
<th>Business model</th>
<th>Business process</th>
<th>Market type</th>
<th>Document-oriented</th>
<th>BP-oriented</th>
<th>XML based</th>
<th>Requirements</th>
<th>Design process</th>
</tr>
</thead>
<tbody>
<tr>
<td>eCo System</td>
<td>Independent; not specific</td>
<td>No</td>
<td>Yes XML-based</td>
<td>No</td>
<td>Yes</td>
<td>eCo server installed on each partner</td>
<td>Implicit bottom-up</td>
</tr>
<tr>
<td>OMG</td>
<td>Independent; not specific</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>CORBA-compliant architecture</td>
<td>Implicit bottom-up</td>
</tr>
<tr>
<td>EBES/ EWOS</td>
<td>Independent; not specific</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Implicit top-down</td>
</tr>
<tr>
<td>EM-RM</td>
<td>Specific to models that follow information, agreement and settlement</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Implicit top-down</td>
</tr>
<tr>
<td>Generic standards</td>
<td>Independent; not specific</td>
<td>No</td>
<td>Yes (e.g., BizTalk)</td>
<td>Yes (e.g., ebXML)</td>
<td>Yes</td>
<td>Standards requirements</td>
<td>ebXML provides an implicit top-down approach</td>
</tr>
<tr>
<td>Specific standards</td>
<td>Models of vertical markets</td>
<td>Yes</td>
<td>Mixed</td>
<td>Some of them (e.g., RosettaNet)</td>
<td>Yes</td>
<td>Standards requirements</td>
<td>No</td>
</tr>
<tr>
<td>Web services</td>
<td>Independent; not specific</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No (may be)</td>
<td>Private of public UDDI</td>
<td>Implicit bottom-up approach</td>
</tr>
<tr>
<td>Agent approach</td>
<td>Independent; not specific</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No (may be)</td>
<td>Agent environment</td>
<td>No</td>
</tr>
<tr>
<td>Workflow approach</td>
<td>Independent; not specific</td>
<td>No</td>
<td>Yes with involvement of end users</td>
<td>No</td>
<td>No (may be)</td>
<td>WfMS architecture-compliant</td>
<td>No</td>
</tr>
<tr>
<td>Brokerage systems</td>
<td>Models involving third party</td>
<td>Yes to a lesser extent</td>
<td>Yes to a lesser extent</td>
<td>No</td>
<td>No (may be)</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>

3.1.3. B2B process

A B2B process is a flexible business process that crosses the boundaries of the organization. It involves sharing information and processes among different businesses (buyer, sellers, manufacturers and third parties) willing to commonly and timely response to business events (e.g., customer order).

Two relevant characteristics of a B2B process are the life cycle and the involved actors (or roles).

Life Cycle. B2B process like any process has a life cycle with different phases in accordance with a business model. It generally includes pre-sale (providing information on goods and services), information stage (finding a match), contract (negotiating and establishing the agreement), supply chain management (fulfillment and shipment), payment (invoice and settlement), and after-sale (satisfaction, providing information). The life cycle may range from minutes to days (or even months) depending on its complexity and the interactions among the actors.

Actors. B2B process involves actors from different businesses. These actors are mostly software (e.g., software components, software agent, legacy systems), but can be individuals or a combination individual-software.

3.1.4. Example of B2B process

Fig. 5 presents an example of a B2B process where a customer order is passed to the sales system. This order triggers the manufacturing system that will trigger an event passed to the parts suppliers systems through the supply system after negotiation and agreement. The parts suppliers’ systems will trigger in their turn their respective raw material suppliers’ systems, to fulfill the customer order, and trigger the customer’s payment. Finally the delivered product will probably be maintained by the aftersale-system.

3.2. B2B requirements

The following requirements will be considered as specifications at each level of ABBA. They are exposed, in this paper, from high level (business process) to low level (implementing and integrating technology).

Requirement 1. A B2B process crosses boundaries of the organization. It concerns with the collaboration between organizational actors (or roles) in different businesses. It is sometimes flexible, i.e., there is no a priori definition of the process, but only a common goal with self-organization of each of the involved businesses.

Requirement 2. A B2B process assumes that each participating business in a B2B EC transaction is autonomous, that is, it is able to decide the conditions of the interactions, i.e., when, how and with whom.

Requirement 3. A B2B process assumes that the market is open, which increases its complexity.

Requirement 4. A B2B process works on the notion of common agreements between businesses, and supports these agreements. Therefore, one person at least must understand the common process that crosses the boundaries of the organization [23]. This person should understand the value chain of each involved business. Hence B2B
process involves an ecosystem where actors interact freely in accordance with their goals and situations.

**Requirement 5.** A B2B process is a sequence of inter-organizational activities. Each activity takes an input from a provider and serves an output to a consumer. Each process has an identity, a pre-condition (trigger state) and post-condition (termination state). This involves different kinds of breakups. Hence, B2B requires interfacing business processes by coordination mechanisms. These are organizational artifacts representing the states of the process. They are required for the continuity of B2B process. However, they don’t have the same weight and priority with respect to the supporting technology. That is, coordination with partners is different from internal coordination. One needs to categorize these artifacts. Such a categorization allows a better understanding of the relevance and the priority of each category in order to decide the appropriate technology. These categories of artifacts are (Fig. 6):

- Artifacts across businesses, i.e., organization business processes and partners’ processes (B-P).
- Artifacts between local primary business processes (P-P).
- Artifacts between local primary and supporting business processes (P-S).
- Artifacts between local business process activities (A-A) when a local business process is decomposed.
- States of business objects as represented in the IS (Information System).

**Requirement 6.** A B2B process may be implemented in a variety of ways, use a wide variety of technological infrastructure (e.g., communication, security, middleware, content management), and operate in a large environment.

Therefore,

**Requirement 7.** A B2B process is characterized by a high degree of automation. This implies that the process is to be supported as much as possible by computer-based and telecommunication-based technology to integrate both processes and data. A B2B integration assumes then that:

- Partners’ value chain are automated and integrated.
- Coordination artifacts should work properly.

Hence, a B2B EC application is based on EAI, EC services, and legacy systems. Moreover, it extends the EAI as shown in Fig. 7. Most partners’ systems and applications will not be altered even if their structures, syntax and semantics are heterogeneous. In addition, businesses’ legacy systems must be preserved and exploited.

---

![Fig. 6. Categories of business processes coordination artifacts.](image-url)
3.3. B2B EC applications and systems

3.3.1. B2B EC application

A B2B EC application aims at providing the right information and processes to the right component (mostly software rather than human involved in B2B process) at the right time to respond to the market events.

A B2B EC application is built on top of current running applications (e.g., order entry, sales, inventory, shipping, purchasing, checking status, invoicing), and data sources (e.g., databases, XML documents, legacy data). This kind of applications consists of sharing the implementation of logic and data among several applications, through the Internet or VAN, with a minimum change to the existing applications or data structures. They consist of implementing middleware intra- and inter- companies to bring together heterogeneous components.

3.3.2. Examples of B2B EC applications

Example 1. The application ‘order tracking status’ would be composed of heterogeneous components and middleware. In this application, a user willing to track the delivery of a parcel, enters the URL into a Web browser (component 1) to request the Web server (component 2) through a protocol (e.g., HTTP) and download a Java Applet (component 3). The user enters then the parcel-tracking number in the field presented by the Applet. The Applet generates and sends a message through a middleware 1 across the Internet. The tracking legacy system (component 4) picks up the message and calls the function to find the parcel’s status. The tracking system connects to the database (component 5) through a middleware 2. Database system returns the result to the tracking system which then routes it through the middleware 1 to the Applet running on the client. Finally, the Applet displays the result to the user.

Example 2. Fig. 8 illustrates a second example where the B2B process specified in Fig. 5 above may be supported by some components of a B2B EC application based upon Web services. This application allows activities such as ‘get quote’, ‘make a purchase’ or ‘track the status’. The manufacturer has a private UDDI where suppliers can register their Web services. The manufacturer’s purchase application looks up a supplier and then sends a SOAP request to get the quote of product. The application informs the buyer of the quotes. The buyer selects a supplier and places an order via a SOAP request. The buyer can also check the status of the order via a Web service provided by the suppliers. This application is built on top of the running applications and databases without altering them.

These two examples demonstrate that a B2B EC application requires resources provided by an environment. We call this environment a B2B EC system.

3.3.3. B2B EC system

A B2B EC system is made up of computational infrastructures and human. The computational infrastructures include heterogeneous hardware platforms and software. Software may be classified into EC services (e.g., Web services, eCo system BIDs), specific software (e.g., help desk, groupware), and legacy systems. These software have to communicate or be integrated using middleware (e.g., transactional, RPC, messaging, distributed objects, Java middleware, message brokers), and communication tools, protocols and standards (e.g., XML over Internet, EDI over VAN). Indeed, an electronically conducted B2B process between businesses is supported by one or more different applications.

Human (e.g., knowledge workers) are involved in the analysis, design, development, deployment, and maintenance of B2B EC applications. As stated before, one person should understand the entire common B2B process to be in charge of it in collaboration with people from partners’ systems.

A B2B EC system is an open system that seeks goals defined by the business. It mainly concerns
with providing the involved businesses' components with the right information (or processes) at the right time. In addition, it must be scalable in order to accept and add new heterogeneous infrastructures.

However, an EC system, or any infrastructure, is not free of charge. It is even very costly. Its related costs should be compared to return on investments. Important elements of these costs are related to the analysis, the development, the deployment and the maintenance. These costs would be streamlined with a methodology including a design process (steps assisted by formalisms, languages, models and tools to assist decision-makers, knowledge workers and technologists in each step). Indeed, every step in EC must be decided. This methodology does not exist naturally, it is conceived from a comprehensive architecture, which is the purpose of the next two sections.

4. Architecture of designing B2B applications

In the light of: (i) the study of some representative architectures, reference models, standards and languages, (ii) the definition of main concepts of B2B EC, and (iii) the specification of the desired properties and requirements of this growing category of EC, we propose ABBA, a comprehensive architecture which aims mainly at deriving an efficient design process (towards a methodology) for deploying B2B EC applications.
4.1. Purpose and objectives

The main objectives of the ABBA, as stated in the Section 1, are:
- Independence business process/technology.
- Guidelines for a design process (towards a methodology) to deploying B2B EC applications.
- Communication between people involved in the design process.

4.2. ABBA components

ABBA consists of four abstraction levels and the interfaces between them (Fig. 9). ABBA is a layered architecture in the sense that each upper level is based upon the specification of the level below it. That is, technology (level 4) implements the EC services (level 3) which support the distributed decompositions (level 2) of the common business process (level 1). Each level is assisted by models as shown in Table 2. Table 3 summarizes the components (levels and interfaces) of ABBA.

4.2.1. Level 1: Business model and business process

This level considers the representation of the business process is derived from a business model. The representation aims at capturing the relevant properties of the business process (independently from the implementing technology). These properties concern with business model and strategy, business process objectives, and the requirements.

Business model and strategy. A business process is derived from a business model and a marketing strategy. The knowledge of the business model and the company strategy are critical to specify the business process [36]. Indeed, a business model and a strategy determine the architecture for the products, services, the elements of the value chain (processes), and consequently the information

Fig. 9. ABBA levels and interfaces.
flows including the description of the various partners and their respective roles [35].

Business process. Once the business model and the strategy are identified, a systematic approach to identify the entire business process can be based upon the value chain elements related to the business model and strategy. For instance, the business process related to e-procurement model [4,21] will differ from e-auction or virtual communities model in many aspects, specifically the required degrees of integration and innovation [35]. The related business model allows the identification of the primary as well as the supporting sub-processes, the flows of the information and processes, and especially the barriers to these flows that require coordination mechanisms. The specification of the entire B2B process requires the specification of the business process of each involved business. This includes:

- The activities of the business process. For each activities, the actions with their pre-condition, post-condition, input and output information.
- The coordination mechanisms and tools.
- The degree of automation of each phase and the supporting information systems. An information system describes the structured data (e.g., database) as well as semi-structured data (e.g., document), and the supporting applications.

4.2.2. Level 2: decomposition and distribution

This level concerns with the representation of the B2B process decomposition and distribution
over the involved actors or roles. It has several objectives:

- With respect to the architecture, the business process level represents the entire business process (crossing the boundaries of the organization). This business process is an abstraction, not really an implementation. The actual running business process consists of several heterogeneous and autonomous sub-processes, and information (or processes) flowing between them. That is, each involved business is concerned with only some sub-processes. Each sub-process is considered here as a view of the entire business process.

- With respect to the work organization, this level copes with the natural distribution of the business process across and outside the organization.

- With respect to EC, the services (or applications) generally focus only on a part of the business process. Mastering the decomposition and the distribution will facilitate a matchmaking between the sub-processes and the built-in EC services in order to reuse them.

- With respect to the implementation (namely the optimization), the decompositions can be seen as instantiations of classes (or templates). In this case, the object-orientation pillars can be used; inheritance to reuse or compose sub-processes, and polymorphism to allow sub-processes accepting different types of input.

4.2.3. Interface 1: mapping and validating decompositions/business process

This interface is a kind of ‘garde-fou’ to make sure that the views (partners’ sub-processes) are represented within the entire business process, and vice-versa the entire business process is commonly understood. It allows the deconstruction and reconstruction of the business process. The deconstruction is the ability to implement the common abstract business as a set of running sub-processes. This mainly involves the implementation of the coordination artifacts (Fig. 6.) to avoid any barrier to the flows of information and processes. The reconstruction is an inverse re-engineering process. It allows the composition of the common business process from its distributed elements. This involves an abstraction of the running sub-processes and their coordination artifacts.

The mapping and validation assume that the business process and its decompositions (as views) are described separately within two distinct levels. The validation makes sure that all the views described separately are within the entire business process, and all the views defined in the common business process do exist really.

This interface would be easy to automate if the decomposition and business process levels are described (or represented) with the same language (or model).

4.2.4. Level 3: EC services

EC services level represents the composition of the services and applications required to support the business process decompositions and their flows. This level consists of a description of the EC services and applications by insulating the interface from the implementation and focusing only on the interface. This description includes (Fig. 10):

- Service naming and description (e.g., language, standard).
- Service type (e.g., EC service, application, database, legacy system).
- Interface:
  - Service-related attributes.
  - Operations (input information, output information).
  - Business objects queried or updated (e.g., invoice, product, order)
- Sub-processes supported by the services.

Fig. 10 shows the relationship between sub-processes and EC services. A service may support many sub-processes, and a sub-process may be supported by many services. Moreover, a service may be based on other services. Table 4 shows some instantiations of this relationship type. These
instantiations are kept in a catalog to further help a matchmaking (interface 2) between business process (or its sub-processes) and EC services.

4.2.5. Interface 2: discovery and matchmaking EC services/sub-processes

This interface deals with discovery and matchmaking between the business process (or its sub-processes) and the potential supporting EC services. Four elements are required for this interface to perform a matchmaking:
1. The validated common business process.
2. The decompositions (sub-processes) of the business process.
3. A private or public catalog of EC services.
4. A browser to discover EC services from a catalog (kind of UDDI) or over the Web.

4.2.6. Level 4: technology

This level consists of the infrastructures of B2B EC system. It represents technological infrastructures that can support B2B EC applications. It is made up of information technology and techniques. It mainly consists of middleware of different types, ERP systems, EAI tools, B2B integration tools, EC services, IS (databases, applications and legacy systems), collaborative tools and hardware platforms. IT department’s resources serve this level.

5. Design process

This section explicitly details the design process (Fig. 11) in accordance with ABBA.

5.1. Steps of the design process

The design process consists of the following steps which are summarized in Table 5.
Step 2: Decomposition and Distribution Specification.
Step 3: Mapping and Validation.
Step 4: Supporting Services and Components Specification.
Step 6: Implementation and Integration Technology.

5.2. Detail of the steps

This section describes step-by-step the design process depicted above.

5.2.1. Step 1: business model, strategy and the consequent business processes

This step is responsible for:
1. Specifying the business model, the strategy and the objectives of deploying B2B EC applications. A B2B process is a derivative from a...
business model, a strategy and a set of objectives. The objectives may change from strategy to strategy. For instance the objective of e-procurement business are total procurement costs, system responsiveness, user satisfaction and process quality. These objectives result from the reduction of the transaction cycle time, the reduction of inventory and the price [4]. A business may have several combined objectives. Well-specified objectives will drive the process design (life cycle) in accordance with those objectives. For instance, to empower relationship with partners, business process must focus on specifying the pre- and post-transaction.

(2) Specifying the consequent business process and naming of the person who is in charge of this business process. The business process specification concerns with:
- The input trigger market event described by information (e.g., customer order).
- The business logic and the exceptions.
- The information flows and the coordination mechanisms.
- The information systems supporting the process (e.g., backend applications, databases and legacy systems).
- The output (digital or physical products or services).

The business process is described with a language. Formal languages (or at least models) are more suitable as communication languages between people involved in the design process (e.g., decision-makers, analysts, designers and
developers). We distinguish at this abstraction level several types of languages for expressing static as well as dynamic aspects. For instance, DFD, Workflow, UMM, UML, Petri Net, State Chart tools (a. k. a. business modeling tools) can be used to express the dynamic aspect of the process, i.e., the life cycle [20]. Entity-Relationship is generally used to describe the static aspects, i.e., the metadata of the stored and exchanged information. However, XML representation of the exchanged information and process is preferable in order to consider further distribution and interoperability problems where current standards (e.g., eCo Systems, Web services) will be very helpful.

Modeling a B2B process as an abstract business process that spans multiple companies which are responsible of implementing the abstraction (Fig. 12) provides many advantages including:

- The ability to create a common business process between companies to automate the integration of their information systems in order to timely react to market events.
- The ability to monitor the business process including the scalability and the openness.

Table 5
Summary of the design process

<table>
<thead>
<tr>
<th>Steps</th>
<th>Objectives</th>
<th>Input</th>
<th>Output</th>
<th>Models/tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>Specification of the services and components that support the business process or its decompositions.</td>
<td>Validated common business process. Catalog of services.</td>
<td>Required services to support business process or its decompositions.</td>
<td>Catalog of services. Browsers, UDDI, Tool to specify services.</td>
</tr>
</tbody>
</table>

Fig. 12. The entire abstract business process and its implementation.
The ability to reengineer the business process at any given time in order to make it more efficient.

The ability to hide the complexity of the local applications upon which the business process is built.

The entire business process should be under the control of one person (the business process administrator: BPA) who understands all the elements of the process. Indeed, the business process (crossing the boundaries of an organization) is neither under the control of the organization (responsible of deploying B2B EC application), nor any other partner’s organization. Making the entire process understood by at least one person becomes then critical.

5.2.2. Step 2: decompositions description

The decomposition specification captures the relevant properties of each view of the business process. These properties concern with:

- An interface consisting of:
  - Services (operations) of the sub-process.
  - For each operation, exchanged information (input and output information).

- An implementation
  - Logic of the sub-process.
  - Internal information flows.
  - Actions (each action is described by a precondition to trigger the action, a post-condition, involved actors, information exchanged or produced).

The steps 1 and 2 may be performed in parallel. While, the person who is in charge of the entire business process works on step 1, each of the involved business partners works on step 2 to define his/her views.

It is preferable to use the same formal language (or model) to define (or model) each of the views. For instance, if the business process is described as set of objects, its decompositions should be described as objects. This will make the validation and mapping easier.

5.2.3. Step 3: mapping and validation

This third step of the design process consists of the validation of the business process described in the first step by the views elaborated in the second step. It consists of:

- Making sure that the business process is described and documented in its entirety.
- Making sure that all the views are described and visible in the business process.

The person who is in charge of the common business process is responsible for performing this step. However, the collaboration of the people in charge of each of the partner’s business process is of paramount importance. Hence, this step requires a collaborative graphical tool to view the entire business process and each of its views. The tool should allow de-construction and re-construction of the entire business process.

5.2.4. Step 4: supporting services and software components

Companies and market are abounded with B2B EC services and components. These services aim at supporting some phases of the business process. A matchmaking between sub-process and appropriate EC services is to be decided. The relationship between the sub-processes and EC services (Fig. 10) may be implemented as a catalog (or a library) of services such as:

- Catalogs services. This service allows the presentation of goods and services (e.g., by product, category and subcategory).
- Search services. These services allow a textual search or product attributes search.
- Matchmaking services structure, record and publish partner’s profile. This kind of services consists of:
  - Profile registration services that are required for registering partners’ profiles.
  - Connecting services that present the protocol to connect to the partner.
- Information aggregation services. This kind of services allows internal actors to gather and aggregate information from registered suppliers or customers, in order to support the decision-making. Aggregation can be static or dynamic. Static aggregation concerns with the pre-registered information about partners and products or services. Whereas dynamic concerns with aggregating from different sources.
- Negotiation services. These are protocols used for bi-lateral as well as multi-lateral negotiations on products and services. Negotiations
can be considered as one of the most complex forms of matchmaking.
- Payment and advanced payment services used for secure payment over the Internet. They also include the security services (of information and payment) and services like a letter of credit.
- Customer care services such as order or shipment tracking services.
- Value-added services include analyze of preferences using cross products, suppliers and industries, decision support for product comparisons and product selection, notary services like non-repudiation of offers, orders, contracts, and support for billing, accounting and digital delivery.

This step consists of deciding the EC services and components used to support the sub-processes. Designers and developers might reuse the existing EC services or developing new ones appropriate to the business process and its related views which are already specified in the previous steps. This requires the identification of:
- The specified business process and its views.
- A catalog of existing EC services.
- The EC standards and languages.

The fourth step results in a set of well-specified EC services and components. An EC service is specified by an interface, an implementation and a protocol for invocation. The interface consists of a set of operations, and for each operation input and output information, in particular, format are specified. The implementation consists of the code implementing its logic (which is not visible to the users of the service). The protocol allows invocation of the service over the Internet (e.g., HTTP, SOAP, IIOP).

5.2.5. Step 5: logical architecture of the application
This step consists of deciding the logical architecture of a B2B EC application, i.e., the components of the application and the links between them. A B2B EC application is made up of two main elements:
1. Components involved in the B2B EC application including:
   - EAI.
   - Existing applications, databases and legacy systems.
   - Existing EC services.
   - Components and EC services to develop for supporting business process.
2. Links between components which will be implemented by means of middleware. Middleware are used to support the flows of information and processes between the components.

5.2.6. Step 6: implementation and integration
The implementing and integrating technologies bring together all the EC services implementing the views of the business process. It consists of three necessary successive decisions:

![Fig. 13. Possible implementations of B2B EC applications.](image)
1. Deciding the integration technology at the enterprise level.
2. Deciding the integration technology in each partner level.
3. Deciding the B2B integration after making sure that each business partner has implemented its EAI.

Fig. 13 shows an example of B2B application integration through some registry mechanisms as middleware. Involved applications, legacy systems, databases, services and components (double rectangle) have their interfaces (simple rectangle) registered in public or private registry mechanism (left-right arrows). These interfaces may be specialized (full up arrows) into CORBA-IDL, eCo Systems BIDs, Web services, ebXML, domain specific standard (such as RosettatNet, OBI, OTP, OFX or ICE), or a combination of them. The registry mechanism (circle) that presents the interfaces, is considered as the middleware bringing together all heterogeneous B2B components.

5.3. Design considerations

We should consider the followings when implementing a B2B EC application:

• Documentation. Business processes that exist within the trading partners must be documented. We must understand and document all the sub-processes and data within each partner's system. That is, how each sub-process works, and how the sub-processes are linked to each other.
• Use tools as much as possible to:
  o Graphically represent the entire business process and its sub-processes.
  o Map a decomposition into the business process and vice versa (to isolate the sub-process from the business process).
  o Collaborate between business process administrator and business partners.
  o Show business activities within each sub-process and the flows of information and processes between sub-processes.
  o Simulate the business logic starting with a business event.

6. Conclusion

The goal of this work is to propose a comprehensive architecture for deploying B2B applications and systems. This architecture aims at: (i) making the business process (crossing the boundaries of the organization) independent from the implementation and integration technologies, in order to get a specified agreed-upon business process; and (ii) providing guidelines for an efficient design process to reduce the complexity of this kind of deployment.

For this purpose, current contributions such as architectures, reference models, standards and approaches have been deeply studied and compared with respect to: the existence of a design process for B2B EC applications, the business model dependence and specification, the business process specification and modeling, the market type dependence, the orientation (process-oriented or document-oriented), the use of XML, and the technical requirements of each approach.

Although, the studied models and approaches complement each other with respect to the abstraction levels, they do not propose explicitly an efficient design process for deploying B2B EC applications to reduce their complexity. Components of B2B applications are still deployed on a case-by-case basis. Accordingly, we have proposed and detailed ABBA, an architecture with four interfaced abstraction levels, and a derived six-step design process. ABBA is a new comprehensive business process-oriented architecture for deploying B2B EC applications and systems.

This is an important issue today where the deployment of complex, open and scalable B2B EC applications and systems are critical for any competitive organization looking forward to be a leader (not a follower).

Future work includes:

• Developing models, formalisms, languages and tools to support the steps of the proposed design process.
• Developing collaborative tools to manage the entire business process and its elements such as: generating and visioning a common business process from organization and partner's processes (re-construction), or dynamically...
managing partner’s business processes (adding, modifying and removing) to deal with B2B EC system openness and scalability.

References


