Inf Syst E-Bus Manage (2009) 7:123–142
DOI 10.1007/s10257-008-0082-4

ORIGINAL ARTICLE

From business to software: a B2B survey

Jürgen Dorn · Christoph Grün · Hannes Werthner · Marco Zapletal

Published online: 26 February 2008

Abstract In recent years business-to-business (B2B) e-commerce has been subject to major rethinking. A paradigm shift can be observed from document centric file-based interchange of business information to process-centric and, finally to service-based information exchange. On a business level, a lot of work has been done to capture business models and collaborative business processes of an enterprise; further initiatives address the identification of customer services and the formalization of business service level agreements (SLA). On a lower, i.e., technical level, the focus is on moving towards service-oriented architectures (SOA). These developments promise more flexibility, a market entry at lower costs and an easier IT-alignment to changing market conditions. This explains the overwhelming quantity of specifications and approaches targeting the area of B2B—these approaches are partly competing and overlapping. In this paper we provide a survey of the most promising approaches at both levels and classify them using the Open-edi reference model standardized by ISO. Whereas on the technical level, service-oriented architecture is becoming the predominant approach, on the business level the landscape is more heterogeneous. In this context, we propose—in line with the services science approach—to integrate business modeling with process modeling

J. Dorn (✉) · C. Grün · H. Werthner · M. Zapletal
Electronic Commerce Group, Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
e-mail: juergen.dorn@ec.tuwien.ac.at

C. Grün
e-mail: christoph.gruen@ec.tuwien.ac.at

H. Werthner
e-mail: hannes.werthner@ec.tuwien.ac.at

M. Zapletal
e-mail: marco.zapletal@ec.tuwien.ac.at

in order to make the transformation from business services to Web services more transparent.

Keywords B2B e-commerce · Business modeling · Business process modeling · Service-orientation

1 Introduction

Traditional approaches in B2B e-commerce followed a strong document-centric approach. Business partners were required to agree on business document standards in order to conduct Electronic Data Interchange (EDI). However, EDI standardization efforts resulted in unambiguous and overloaded business document types. Conducting EDI required again the need for agreements on subsets of the standards between business partners. Thus, the implementation and operation of traditional EDI systems—often communicating over special networks—were costly. As a consequence, the participation in B2B e-commerce was reserved to larger enterprises that were able to afford the costs.

Since then research has moved towards process-centric B2B approaches. These approaches capture the flow of business information between business partners, which is exchanged to reach a certain business goal. Business process models capture the business information that is required in each step of a collaborative business process. This eliminates redundantly transmitted information, which in turn lowers the risk of semantic differences between exchanged business information. In addition, business processes might be subject to standardization efforts in the future, which allow for a cost-effective implementation of commercial off-the-shelf software (COTS) supporting these standardized processes. Which business goals shall be reached by business processes is represented in business models. They formalize the economic rationale by modeling the values exchanged between business units where values are typically differentiated into products, services or money.

Regarding software development, we observe a major paradigm shift towards service-based communication—known as service-oriented architectures (SOA). The SOA concept promises reuse of components and allows for an easier implementation of communication across heterogeneous platforms and among enterprises based on open and free specifications. Furthermore, SOA may be parameterized by deployment artifacts such as machine-readable business process specifications, workflow descriptions, and business document schemes. Such deployment artifacts enable a more flexible and easier adoption of service-based systems to changing environments—even at runtime.

Novel B2B approaches combine the two “paradigm” shifts on business and technical level. Technology independent specifications on a business level are employed to infer artifacts on the implementation level. Moreover, knowledge captured on the business level allows deducing artifacts for different technical environments (i.e., platforms). This facilitates enterprises to adapt to changing and newly emerging technologies. This is in line with the services science approach

(Chesbrough and Spohrer 2006) focusing at service driven innovation. But one should note that services are defined differently in management science and in computer science. In the latter one, a service is a simple or complex task executed within an organization on behalf of a customer (O'Sullivan et al. 2000). In management science, a service is defined as a business economic activity (mostly intangible in nature), offered by one party to another in order to achieve a certain benefit (Zeithaml et al. 2005) (Kotler and Keller 2005) and "generated" by business processes. In this context it is interesting to note, that modeling of business services is not so widespread propagated as for example the modeling of business processes.

Flexible service design and implementation requires a (semi-) automated path from business models down to business process models and finally, to deployment artifacts (Margaria and Steffen 2006). Hence, these layers must be linked in order to derive artifacts at a lower layer from formal descriptions at a higher layer. Furthermore, there should be a support also into the opposite direction. When new services become available on a technical level (e.g., a new booking service, or a new encryption algorithm), they should be transformable into business services and easy integrable into existing business models. Whether a new service replaces an existing one depends on business objectives. In essence, the objective is to abstract technical implementation details from business users (Brodie et al. 2005).

In this context the paper provides two contributions: (1) It gives an overview and a classification of the most prominent B2B approaches. We show both, business- and implementation-related specifications and classify them in terms of the Open-edi reference model (ISO 2004). We discuss advantages and drawbacks, outline overlaps and sketch their roots and progressions. (2) We discuss a conceptual path from business models down to deployment artifacts for SOA environments.

The remainder of this paper is structured as follows: Sect. 2 introduces the Open-edi reference model and the refinements we made for our paper. In Sect. 3 we outline business-related approaches and in Sect. 4 we review the most promising SOA related specifications. Section 5 provides a conceptual design of a (semi-) automated transformation starting from business models via business process models to Web services, enriched by semantic enhancement. Section 6 concludes the paper.

2 Refining the Open-edi reference model

The Open-edi reference model (ISO 2004)—standardized by the International Organization of Standardization (ISO)—groups EDI related standards into two categories (cf. Fig. 1 for a slightly modified version). The *business operational view* (BOV) addresses the semantics of electronic business, hence the semantics of business collaborations and related business information exchanges. Specifications going into the BOV capture business knowledge in a technology independent way. We also refer to such technology independent specifications as methodologies. The *functional service view* (FSV) addresses the technologies and the implementation aspects to support business collaborations specified in terms of BOV related specifications. In other words, technologies on a lower level implement higher level

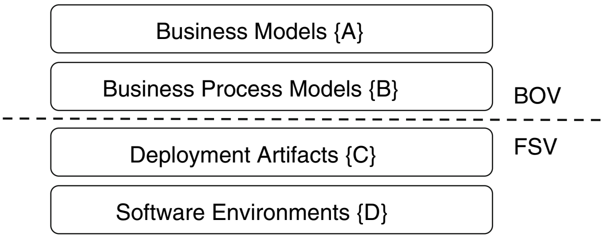

Fig. 1 Classification schema based on refinements of the Open-edi reference model

methodologies. Since the BOV describes a business in a technology independent way, different FSV implementations (i.e., deployments for different platforms) may be derived.

In order to classify existing approaches in the field of B2B, we differentiate between business models, business process models, and deployment artifacts. For this we refine the Open-edi model (see Fig. 1) as follows: The BOV is split up into *business models* {A} and *business process models* {B}. In this paper, we do not elaborate information modeling being part of business process models. The FSV comprises specifications for *deployment artifacts* {C} as well as *software environments* {D}. Due to space limitations, we do not discuss software environments in this paper. Figure 2 gives an overview of the business- and implementation-related specifications that are discussed in the Sects. 3 and 4, respectively. The references in braces in Fig. 2 are used as pointers to the description of each specification.

3 Business-related specifications

On the upper layer, we differentiate between business models and business process models. Business models describe the exchange of values (goods, services and money) between business partners on an abstract level with the overall goal to generate benefit for each participant. Business process models are located on the next lower layer. They describe the relationships between the partners by making assertions about the flow of information and type of interaction. From a generic point of view, a business model therefore defines the *what* value while a business process model defines *how* the value is created (Bergholtz et al. 2002).

3.1 Business models

In the definition of business models, we follow Timmer (Timmers 1998), who defines it as an architecture for the product, service and information flows, including a description of the various actors and their roles, together with a description of the sources of revenues and potential benefits. Several other definitions can be found in

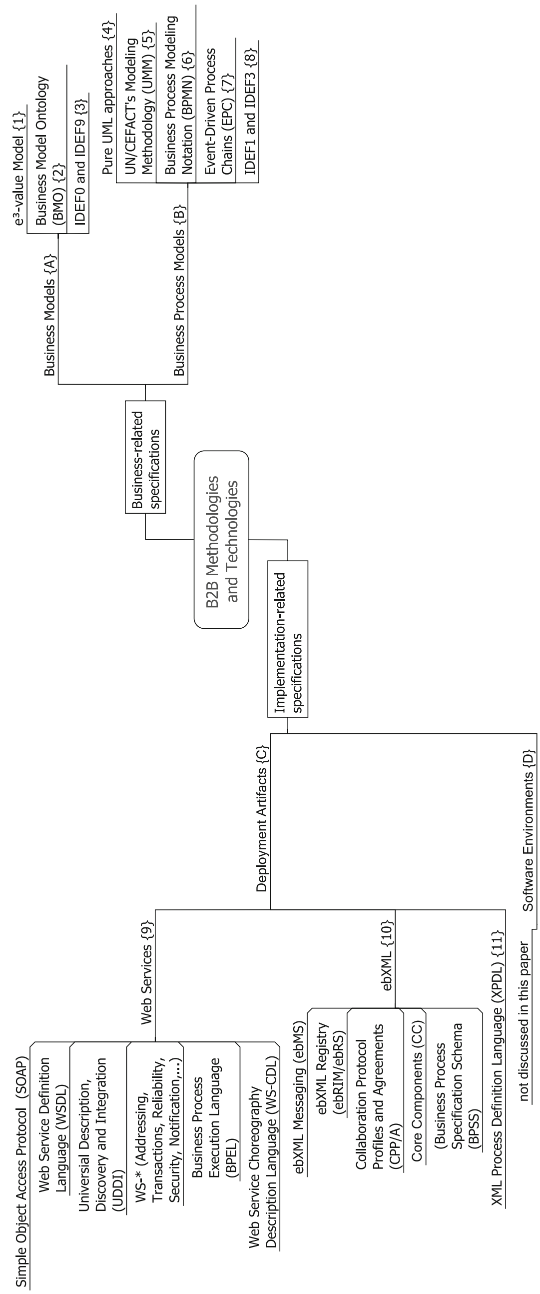

Fig. 2 Overview of business- and implementation related B2B specifications

(Pateli and Giaglis 2003) that presents a framework for structuring business models into six sub-domains, with the sub-domain definitions forming an individual part.

Over time the focus of business model research changed (Gordijn et al. 2005), ranging from establishing taxonomies of business models, to describing elements of business models, and finally to building business model ontologies. Such ontologies facilitate stakeholders to establish a common understanding by providing a set of vocabularies and concepts that is used to describe the business logic. In fast moving market conditions with the entrance of new players and a deconstruction of the value chain, stakeholders are supposed to form business networks in a flexible way on a plug & play basis. These circumstances demand business models to be expressed formally so that they can be processed in a machine-readable way: (a) this ensures that they can be easily adapted to changing requirements; (b) they can be analyzed by tools that are capable of simulating different business scenarios to facilitate the selection of the most sustainable one.

Different approaches to design business models can be found, e.g., being based on the REA (Resources, Events and Agents) ontology (McCarthy 1982) or by building a business object ontology from the e-Business Process Handbook of MIT (Terai et al. 2003). If business models are specified formally, the actors (customers and enterprises) and corresponding relationships, the exchange of value objects (products, services, money) and the business activities to create the values are modeled.

Two promising and mature business model approaches are the e^3-value and the Business Model Ontology (BMO), which are illustrated in short in the following.

3.1.1 e^3-value model ({1} in Fig. 2)

In e^3-value (Gordijn and Akkermans 2001, 2003a, b) a business model is regarded as a value constellation, i.e., a network of enterprises that jointly create and distribute objects of economic value to satisfy a consumer need. Focus is on an economic value proposition, i.e., expressing the objects of values an actor is willing to exchange for other objects. The model ensures the concept of economic reciprocity, i.e., if an actor delivers a good so that he or she gets another good in return. Hence, the model illustrates which actors have economic transactions with each other on an abstract level, without the internal processes necessary to create these values. Emphasis is on showing who is doing business with whom. Actors might decide to bundle their products or to appear as virtual enterprises. The e^3-value approach supports the representation of this typical form of e-businesses. Use Case Maps (UCMs) are used to outline the path via which objects of value need to be exchanged in reaction to a consumer need. Moreover, it can be combined with goal-oriented modeling (Gordijn et al. 2006) by mapping strategic business goals of the business actors to the value model, which subsequently shows the value exchanges necessary to realize the business goals. The feasibility of a business model may be evaluated by means of profit sheets and sensitive analysis.

3.1.2 Business Model Ontology (BMO) {2}

In BMO (Osterwalder and Pigneur 2002), business models are described based on four perspectives, comprising product innovation, infrastructure management, customer relationship and financial aspects and the relationships between them. In contrast to the previous mentioned e^3-value model, which describes the network constellation as a whole, this ontology rather focuses on a specific actor and outlines his position in the business network and how he makes profit. In this way, the perspective *product innovation* mainly refers to the value the actor offers to a specific target customer segment whereas the perspective *customer relationship* refers to the different distribution channels to deliver the created value. The perspective *infrastructure management* outlines how this value is created by regarding the resources of the actor as well as his partner network. The perspective *financial aspect* is influenced by all these three elements and determines the actor's profit model.

In (Gordijn et al. 2005), the two business modeling approaches—e^3-value and BMO—are compared based on a framework to identify common characteristics as well as differences, with the overall goal to integrate them in future. Both approaches use graphical modeling tools to visualize their business models and to facilitate the understanding of the model. The main difference between the two approaches is their focus. The e^3-value focuses on the network of business partners and exchange of values. BMO picks out a specific enterprise and describes its business interactions by pointing out offering- and customer-related aspects. As the authors of (Gordijn et al. 2005) state, the integration of different ontologies seems to be a promising way of combining the advantages of different models.

3.2 Business process models

According to Hammer and Champy a business process is defined as a flow of related activities that together create a (customer) value (i.e., a service or a product) (Hammer and Champy 2001). A business process model is an abstract description of the flow of one or more business processes containing activities and required resources.

3.2.1 Pure UML approaches {4}

Although UML was initially introduced as a modeling language for object oriented software systems, its flexibility and extensibility have attracted business modelers and analysts. In order to describe the behavior of a business process, UML activity diagrams might be used. For specifying interactions between participants on a lower level, one might also utilize sequence diagrams. The semantics of activity diagrams were significantly changed from a specialization of state machines in UML 1.x (Object Management Group 2005) to rather petri-net like semantics in UML 2 (Object Management Group 2007). Nevertheless, activity diagrams of both major UML versions are adopted for business process modeling.

In (Kim 2002), Kim suggests a UML 1.x based approach for modeling B2B processes. The proposal comprises activity diagrams for modeling collaborative processes as a flow of transactions. A transaction is denoted as an activity and implies a message exchange between two business partners. The flow of business documents within a transaction is further decomposed using sequence diagrams.

In (Kramler et al. 2005), the authors propose a UML 2-based and platform independent approach for modeling collaborations between Web services. The paper identifies relevant diagrams and concepts of UML to model collaborations starting from a high abstraction level to fine-grained service interactions. The different levels of granularity of the approach are compared to the layered architecture of the eCo framework (eCoWorking Group 1999). The technique argues to incorporate the main concepts of the Business Process Specification Schema (BPSS) and the Business Process Execution Language (BPEL) that facilitate a mapping to these types of deployment artifacts.

In (Gardner 2003), a UML profile for BPEL is proposed. The profile provides a direct mapping to corresponding BPEL constructs. However, this technique offers only a graphical representation. Furthermore, the approach is rather generic and does not address the specific needs of B2B.

All three approaches outlined offer means to model collaborations between entities and to derive deployment artifacts thereof. However, they define essentially graphical abstractions and have shortcomings with respect to B2B process modeling. Firstly, none of them includes requirements elicitation in early development stages. Secondly, the scope of these approaches does not consider the reuse of artifacts. But, the reuse or even standardization of processes and business information can be seen as key for the acceptance of process- and service-based e-business technologies. A common and standardized set of collaborative business processes would foster the development of standard e-business software (i.e., COTS) and would ease agreements between business partners. Thirdly, these are rather academic research approaches, which up to now are not widely accepted by the industry.

3.2.2 UN/CEFACT's Modeling Methodology (UMM) {5}

UMM is a UML-based methodology, defined as a UML 1.4.2 profile (UN/CEFACT TMG 2006)—including stereotypes, tagged values and constraints. UMM is a standardized methodology developed by UN/CEFACT and well accepted in the field of B2B modeling. In its early stages, RosettaNet (RosettaNet 2002) contributed experiences from its application in the IT, telecommunication, and semiconductor industry to the development of the UMM. Furthermore, UMM has been the modeling methodology of choice within the ebXML framework. But one should note that UMM was at no time a direct part of ebXML, which comprises solely FSV specifications.

UMM consists of three views—the business domain view, the business requirements view and the business transaction view. In the *business domain view*, the business analyst gathers existing domain knowledge from stakeholders.

Furthermore, the business domain view might be used to classify the domain under consideration (and the business processes existing therein) according to a domain ontology. This domain ontology might correspond to UN/CEFACT's Common Business Process Catalog (CBPC) (UN/CEFACT TBG14 2003) or Porter's Value Chain (PVC) (Porter 1979), but might also be self defined. The *business requirements view* elicits requirements of desired and to-be-designed business collaborations. Furthermore, the business analyst might detail existing processes in order to reuse or incorporate parts thereof. The last view—the *business transaction view*—describes business collaborations and business transactions. Business collaborations are conducted between two or more partners and define a flow of business transactions. The concept of a business transaction represents an information exchange between exactly two business partners.

Mappings of UMM to deployment artifacts like BPSS (Hofreiter et al. 2006a) and BPEL (Hofreiter and Huemer 2004) are proposed. Academic tool implementations exist that generate these types of process specifications based on the proposed transformation rules (Ilger and Zapletal 2006; Liegl et al. 2006). Furthermore, the UMM meta model has been designed for the reuse of business collaboration models or parts thereof. UN/CEFACT targets at building and maintaining a registry of reusable business artifacts—including business collaborations, business transactions and business information. A mapping of UMM artifacts to the ebXML registry meta model is suggested in (Hofreiter et al. 2006b).

Nevertheless, UMM has also shortcomings. Currently, there is a lack of tool support by software vendors. Criticism targets also at the complexity of UMM and its meta model. This results from the fact that the methodology provides means to capture complex collaborations with clearly defined business transaction semantics. In addition, its heavy focus on reuse contributed to UMM's intricacy.

3.2.3 Business Process Modeling Notation (BPMN) {6}

BPMN has been developed by the Business Process Management Initiative (BPMI). Recently, the BPMI contributed their work to the Object Management Group (OMG), which is well known through standardizing UML. As a result of the merger of the business process management activities of those two groups, the BPMN became a final OMG specification (Object Management Group 2006).

BPMN provides a small, but clearly defined notation for modeling business processes. The simple notation enhances the understandability of BPMN diagrams among different groups of users. Since BPMN is defined from scratch as a BPMN, it has a definite business centric approach. According to (Emig et al. 2006), this is a major advantage compared to UML-based approaches, since UML has its roots in object-oriented software design. The authors state that UML is unfamiliar to most business analysts and that UML itself defines no mapping to process specification languages like BPEL. BPMN on the other hand, defines a direct mapping to BPEL, which is also part of its specification. The closely defined relationship between BPMN and BPEL should narrow the gap between business process models and process implementations thereof.

BPMN defines Business Process Diagrams (BPD) to capture a business process. A BPD describes the flow of a process using flowchart techniques. The modeled process might be either internal to a company (private process) or collaborative if executed between two or more participants (public process). Furthermore, BPMN allows to model the interface that a private process exposes to its outside world. The interface of such a process defines what message exchanges are required in order to interact with it. Process interfaces are called abstract processes.

In the field of B2B, we focus on collaborative BPD's. Unfortunately, BPMN specifies mappings to BPEL only for internal processes, but defines no complementary generation of BPEL artifacts for collaborative processes. According to the BPMN specification, BPSS is considered as a target language for collaborative BPD's in the future.

BPMN provides—as indicated by its name—only a notation for capturing business processes. However, in order to capture the complexity of a B2B process a supporting methodology guiding the business analyst from the requirements of the business domain to the formal business collaboration specification is inevitable. This is not provided by BPMN. Lastly, we note that BPMN has no focus on reusing artifacts. There is currently no approach known to us that covers a business registry binding of BPMN artifacts.

3.2.4 Event-Driven Process Chains (EPC) {7}

EPCs are a process-oriented modeling technique proposed by Keller et al. in (Keller et al. 1992). Since this method is used for the definition of business processes in SAP/R3 and other ERP systems, it gained a great attention in companies worldwide. ARIS is a tool set that supports besides other modeling approaches the EPC approach and is continuously extended to support recent developments in the IT-world (Scheer et al. 2006). The modeling approach is based on a sequence of events and functions (activities) that constitute a business process. Logical connectors (logical and, or and xor) enable the description of branching actions and conditions for the execution of parallel activities. Extended EPC support the modeling of resources, data objects, organizational units and services. Further, the linkage between processes is supported. EPCs are considered as a method to be used very easily by domain experts to specify business processes. EPCs provide also bindings to the deployment layer. In terms of the Web service stack, the authors of (Ziemann and Mendling 2005) propose a transformation of EPCs to a BPEL representation.

3.2.5 Integrated Definition Methods (IDEF) {3} and {8}

The IDEF suite contains specifications for business models as well as business process models. However, since we discuss IDEF mainly for process modeling, we present the entire suite at the business process model layer.

IDEF is a suite of modeling languages used especially in the US governmental as well as military sector. The languages are standardized by the US National Institute of Standards (NIST). Its origin was the Structured Analysis and Design Technology (SADT), which is now the first part (IDEF0) of the suite. IDEF0 is used for a functional analysis of systems and is also proposed as a method for analyzing business models (Mayer et al. 1995a). IDEF1 supports the modeling of information objects required by a system and IDEF1X is a method for designing relational databases with a syntax designed to support the semantic constructs necessary for developing a conceptual schema. The IDEF3 process description capture method provides a mechanism for collecting and documenting processes. It captures precedence and causality relationships between situations and events in a form natural to domain experts by providing a structured method for expressing knowledge about how a system, process, or organization works. IDEF4 follows an object-oriented modeling approach, whereas IDEF5 supports the modeling of ontologies. A further extension is IDEF9 (Mayer et al. 1995b), a business constraint discovery method designed to assist in the discovery and analysis of constraints in a business system. Constraints might be policies, rules, conventions, procedures, contracts, agreements, regulations, societal and physical laws that define the structure of an enterprise. The IDEF suite is supported by a set of tools that aims to support business re-engineering in enterprises. SADT is an old and well-accepted specification method especially in military and real-time systems engineering.

Although object-oriented analysis and design are introduced in IDEF, in most industries UML has replaced SADT so that the IDEF suite is only applied in some niches.

4 Implementation-related specifications

The technical layer corresponds to the FSV of the Open-edi model, refined into the *deployment artifacts* layer and the *software environments* layer. Deployment artifacts comprise business process specifications, workflow descriptions or document schemes in a machine-processable language. The software environments layer corresponds to concrete implementations of information systems. Software environments use deployment artifacts in order to execute or participate in a certain process—e.g., a workflow engine executing a BPEL process. Due to space limitations, we do not discuss the latter.

4.1 Deployment artifacts

4.1.1 Web services {9}

In its beginning, the Web service framework was based on three specifications: Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL), and Universal Description, Discovery and Integration (UDDI). WSDL describes the interface of a Web service. In other words, it specifies what a service provides together with the parameters it consumes and its return value. Based on a WSDL description one interacts with a Web service using SOAP messages. UDDI specifies a repository to store and retrieve specifications (e.g., WSDL descriptions) in order to interact with a certain service. SOAP and WSDL are both standardized by the World Wide Web Consortium (W3C) and UDDI is under standardization of the Organization of the Advancement of Structured Information Standards (OASIS).

Web services realize a platform-independent and loosely coupled communication based on open and free standards. Hence, they provide means to connect heterogeneous systems or to integrate legacy systems. Both, industry and academic research have soon discovered the potential of Web services. Consequently, the Web service framework has significantly grown in a bottom-up manner. Taking primarily SOAP and WSDL as a base, a lot of specifications and approaches have been built upon these standards in order to realize more specific requirements. Specifications for addressing services (WS-Addressing), reliable messaging (WS-Reliable Messaging), security (WS-Security), or for realizing transaction mechanisms (WS-Transactions) have emerged, just to name a few. Due to space limitations we will not elaborate the multitude of WS-* specification, but rather focus on process specifications.

In order to implement a business process using Web service technology we need to map the flow of a business process to a set of Web service interactions. Two major specifications for defining business processes on the Web service layer emerged in the past—the Business Process Execution Language (BPEL) (OASIS 2007) and the Web Service Choreography Description Language (WS-CDL) (W3C 2005b).

BPEL describes a process by a flow of activities representing interactions of Web services. In general, activities specify the sending or receiving of a message to/from a service. Service interactions are further specified by references to WSDL descriptions. A BPEL process might either be an abstract process - also called business protocol—or an executable process. In terms of BPEL, an abstract process defines the same behavior as an executable process but abstracts data handling.

BPEL has its origins in the orchestration of Web services by describing a process from a specific partner's point of view. Although, the definitions of orchestration and choreography are often not clearly separated, we emphasize on a distinction between the two terms. In terms of Web services, an orchestration is comparable to a composition of Web services.

onship between service A and the services B and C. Furthermore, whoever calls A neither have an influence on the orchestration nor is the orchestration visible in most cases. A choreography, however, describes an interaction between two or more Web services, whereby each participating service behaves as a peer. There is no center of control like in an orchestration.

Since BPEL describes the behavior of a specific partner's service interface, the BPEL descriptions of each participant of a choreography must be complementary. Nevertheless, if each one describes the same process from his own view in isolation, the respective BPEL descriptions do not match. Thus, a collaborative process must be specified on the layer of a business process model in order to derive complementary BPEL descriptions for each participant.

The WS-CDL is the current effort of the W3C to standardize a choreography language. It describes a collaboration between two or more peers from a global point of view by capturing the sequence of message exchanges. An agreed choreography description serves as a kind of contract between all participants of a process in order to achieve their respective business goals. Each participant is required to implement its part of the process according to the agreed choreography description. Partner specific abstract process specifications might be derived from the global choreography to facilitate local implementations—e.g., deriving BPEL stubs from WS-CDL as proposed in (Mendling and Hafner 2005). Furthermore, the choreography description might be used at design time to check the compliance of a local service implementation and at run time to determine the current state of a choreography. Hence, one can calculate the next steps in the process and also track failures and exceptions. WS-CDL has a heavy focus on reuse, i.e., the same choreography description might be reused in different geographical and industry contexts and new choreographies might be composed of existing ones.

4.1.2 ebXML {10}

ebXML was a joint initiative between the two standardization organizations UN/CEFACT and OASIS. Its clear focus lays on providing a technical environment for B2B collaborations meeting the specific requirements of B2B. The framework incorporates the concepts of SOA and is the second important approach in this field beside Web services. The ebXML framework provides a set of five specifications: Messaging (ebMS), Registry (ebRIM/ebRS), Collaboration Protocol Profiles and Agreements (CPP/A), Core Components (CC) and BPSS.

The ebXML messaging (OASIS 2006) is defined on top of the SOAP with Attachments (SwA) specification. The SOAP message itself contains technical information for the respective message handlers concerning routing, security, correlation, and reliability, just to name the most important.

The idea of ebXML registries is to provide standardized repositories for managing B2B related content. Such content includes, but is not limited to, business partner profiles and standardized business process descriptions and business documents. In a B2B scenario, registries allow business partners to find each other electronically. The ebXML registry specification is divided into two parts. The registry meta model, covered by the registry information model specification (ebRIM) (OASIS 2005a), defines how actual content is associated with meta data and classifications. The interface a registry has to provide for communication with the outside world as well as internal services are defined within the registry services specification (ebRS) (OASIS, 2005b). Internal services a registry must implement are, for example, lifecycle management, versioning, content cataloging and validation and access control mechanisms.

Collaboration protocol profiles (CPP) (OASIS 2002) allow partners to specify their party information and most notably their capabilities in terms of conducting electronic business. The capabilities denote which business processes are supported in which role and by which technical infrastructure. Furthermore, a CPP covers technical parameters in respect to the execution of a certain business process. This includes protocol and endpoint definitions as well as time-outs and security requirements. A collaboration protocol agreement (CPA) (OASIS 2002) captures an agreement between two business partners on a certain business process under covenant technical parameters.

Core components (CC) (UN/CEFACTTMG 2003a) represent building blocks of reusable business information. The approach heavily relies on the principles of context. This means, that a core component represents business information that is free of context, but applicable across many business domains (e.g., the concept of person). If the core component is used in a certain business domain the corresponding context is applied and the block of information is now called business information entity (BIE). An actual business document used in a certain business domain is assembled as a set of BIE's.

Business process specification schemes (BPSS) (UN/CEFACT TMG 2003b) capture the choreographies of collaborative business processes in a machine-interpretable manner. BPSS process specifications are means for configuring e-business systems at runtime in order to execute a certain business collaboration. ebXML does not mandate an approach to create BPSS process specifications, but recommends using UMM for collaborative process modeling.

Nowadays, five years after the ebXML initiative we attempt a short synopsis. In our opinion, the ebXML framework provides a sophisticated approach for dealing with the specifics of B2B. At the time of its standardization in 1999 until 2001 and partially even today, some parts of ebXML are superior to Web services in respect to B2B: ebXML messaging was designed from the beginning with security and reliability in mind. In case of Web services, some partly overlapping specifications addressing security and reliability on top of SOAP have been introduced in the past years. In the field of registries the ebXML registry meta model and the registry services are more advanced than UDDI. Profiles and agreements are currently not considered in the Web service specification stack. In order to represent document structures Web services rely on XML schema, which is a well-accepted specification. However, there is only little consideration of information reuse in the field of Web services, on the contrary to ebXML CC which have been designed with reuse in mind. In the field of choreography description languages, WS-CDL is the first specification that is somehow competitive with BPSS. However, one should note that the initiative has failed in terms of its aims. Today, the use of ebXML in comparison to Web services is obviously low, which we accredit to poor vendor support.

4.1.3 XML Process Definition Language (XPDL) {11}

XPDL is a graph-based language standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products like modeling and simulation tools as well as workflow engines (WfMC 2005). XPDL 2.0 has been closely aligned to BPMN in order to become the interchange format of choice for BPMN processes.

In XPDL, a process is composed of a set of activities that may be connected by transitions. An activity may be connected at its input side with different transitions in a join element and at its output side with several transitions in a split element. Four kinds of activities are distinguished: a generic action, a route activity (with an empty activity body), a block activity (containing a set of activities) and a sub-flow activity that calls another process. Activities may have attributes such as a deadline or a priority. Furthermore, a performer (either an application or a human) can be referenced. A transition is used to connect activities and to express conditions whether certain activities shall be executed. A data field element can be used to pass data between activities and transitions. If a Web service shall be controlled by XPDL processes, the extended attributes element of an activity must be used since no immediate integration is standardized.

5 From business models to deployment artifacts

Flexible service design and implementation require a path from business models down to business process models and finally, to deployment artifacts. This is expressed by Fig. 3, where such a possible path is described on a conceptual level, re-using the distinction of Fig. 1. The upper layer ({A}) corresponds to the business perspective, providing and defining services, organizational units, business rules and resources. It may also include business objectives and corresponding measurement values (e.g., profit or number of customers). Business rules, logically relating such values, describe tactical knowledge how an enterprise may achieve its objectives. A layer below ({B}) business processes will be semi-automatically selected and adapted in order to implement the defined services, considering business rules and objectives. Decision points in the process will typically access the mentioned measurement values. It is assumed that process activities can be implemented by a set of Web services (layers {C} and {D}, respectively). The required Web services, their sequence and used resources may be automatically determined by Web service composition methods. The composition is guided by rules and objectives transparently delivered through the business process layer.

Partners within a network define their business needs and models in order to describe which services and related values they are willing to exchange. Furthermore, they need to specify appropriate business processes that support value exchanges as described in their business model. Both, business models and collaborative business process models should be published and interlinked in business registries. One should note that an automatic generation of a process model, describing a flow of activities, from a business model that defines objectives and requirements, does not seem to be possible. Thus, in order to fulfill a certain business model a manual selection of an appropriate business process model from a given set of business process models is required.

Having agreed on business models and business process models, each participant is then required to configure its local e-business system in order to accomplish its part of the process. In a SOA environment, a system has no hard coded behavior but can be configured using deployment artifacts to achieve the desired behavior. Such deployment artifacts or at least parts thereof can be derived using a collaborative process model. We outlined approaches for a model-based generation in Sect. 3.2 in

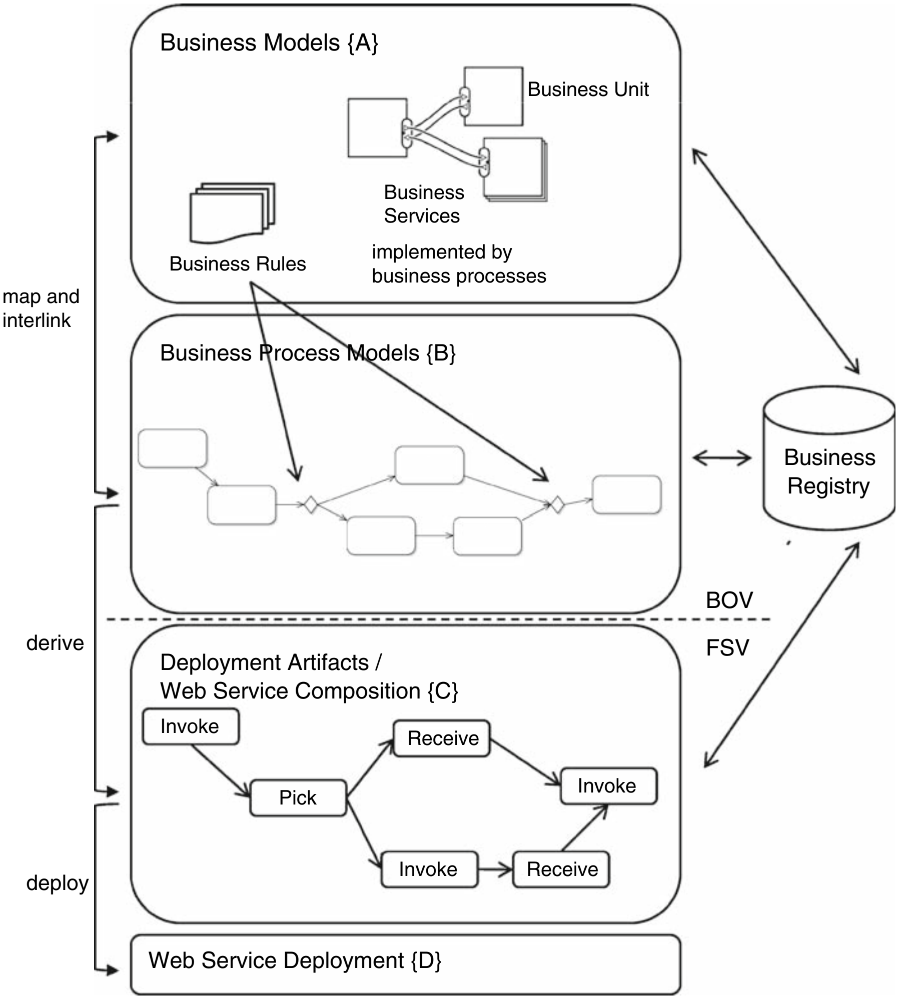

Fig. 3 B2B transformation process: from business models to Web services

terms of the respective process modeling methods. Generated deployment artifacts might include BPSS and WS-CDL choreographies or abstract BPEL processes. In order to foster their reuse in different contexts, these artifacts will also be registered within a business registry. In the following we discuss three issues important in this context: Web service composition, service level agreements (SLA), and semantic-based transformation.

5.1 Web service composition

A crucial issue is to obtain a composed set of Web services given the business process description. A promising approach in this context is to work via the automated composition of intermediate workflows that choreograph a set of Web services. Typically such a Web service composition is achieved by a planning tool that uses the similarity between the WSDL representation of Web services and the representation of actions in the Artificial Intelligence (AI) planning community (Rao and Su 2004). An AI action is composed of a list of preconditions and a list of results. We can interpret input arguments of a Web service as preconditions and the output arguments as results of an action. AI planning methods can be applied to search for a plan (i.e. a number of actions with certain sequence constraints) to solve a certain goal (e.g., a customer receiving a product). In principle, such a solution can be deployed as a workflow. Typically, AI-based planners search backward-oriented, thus first looking for an action that has a result matching the goal. Then the preconditions of the action are investigated and a plan is searched for, which fulfills the preconditions. Applied planners differ in whether only one sequence of actions (Web services) is found or whether also the parallel call of Web services is considered (non-linear planning). Some of these planners can guarantee that a plan is found that contains the minimal number of calls. This is a kind of an optimization strategy, when however, other optimization criteria are more important (e.g., the time required by the whole plan or the costs attributed to the sum of all called Web services), more sophisticated planners are required.

The objective to achieve the full flexibility of SOA requires that we acquire business goals of the business model layer and transfer these goals into measurable values on the deployment/composition layer. This includes also antagonistic goals of participating units. In Rainer (2006) and Dorn et al. (2007), logic programming is applied to the Web service composition problem. It enumerates all feasible solutions and evaluates them according to explicitly expressed goal functions. The system is thus able to deliver a process composed of Web services found within a repository with the best evaluation (Blake et al. 2005). However, the solution is restricted to small to medium size problems since all solutions are considered.

5.2 Service level agreements

In order to compose services, SLA as contracts between a service provider and consumers are important. They define the services provided, the associated metrics, acceptable service level liabilities on both partner's sides and specific action to be taken in certain circumstances. SLAng (Skene et al. 2004) is a formal language for specifying such SLAs. SLAng is motivated by the outsourcing business and supports the composition and monitoring of SLA. (Masche et al. 2006) demand a general modeling of such SLAs for B2B systems, with the expectation of stronger decoupling of service consumer and provider. SML (W3C 2007) is a proposed W3C standard for modeling IT services in the outsourcing industry. In contrast to SLAng, however, it has no service specific features defined. SLAs are also comparable with Quality of Service (QoS) approaches for Web services, however, QoS focuses on technical requirements. In SLAng as well as in QoS modeling business objectives and business attributes (e.g., costs) are not yet addressed.

5.3 Semantic-based transformation

The level of automation in the transformation process as expressed in Fig. 3 could be enhanced by adding semantics to the individual layers. Semantics have to be applied for defining hierarchies of potential business objectives, business services, business processes and business documents. By constraining concepts from these hierarchies against each other (e.g., how is a business object related to a business process) we obtain business ontologies both guiding and constraining the construction of deployment solutions. In addition, computation will involve searching for services based on functional and nonfunctional requirements and inter-operating with those that were selected. However, this approach will not scale and services will not be able to interact automatically without further data, protocol, and process mediation. Hence, machine-processable semantics are critical for SOAs to reach their full potential (Sheth et al. 2006).

As a consequence, semantic Web service frameworks, namely the Web Service Modeling Ontology (WSMO) (DERI 2006), OWL-S (W3C 2004), and WSDL-S (W3C 2005a) are gaining ground. These technologies can be used as a starting point to raise the degree of automation of the outlined transformation process.

6 Conclusion

In this paper we presented an overview of approaches, methodologies, specifications and technologies in B2B e-commerce. We classified them based on the Open-edi reference model, which was refined into the layers: *business models*, *business processes*, *deployment artifacts*, and *software environments*. When creating business networks these four layers have to be addressed in a top-down approach - from defining business model descriptions to deriving abstract business processes into aligning IT-based processes. For each layer we listed approaches both from the industry as well as from the academic field and discussed their advantages and shortcomings.

The goals were to bridge the different views, to classify existing methodologies and technologies in this area, and to indicate how specifications at the different layers fit together. It is apparent, that more research is needed to integrate these different layers, where we highlighted a possible path from the description of business logic in form of business models to the derivation of software artifacts. This illustrates, that the gap between the business and technical perspective of B2B e-commerce is becoming smaller, but still exists.

References

Bergholtz M, Jayaweera P, Johannesson P, Wohed P (2002) Process models and business models—a Unified Framework. In: Proceedings of ER (Workshops). Springer LNCS

Blake MB, Tsui KC, Wombacher A (2005) The EEE-05 Challenge: a new web service discovery and composition competition. In: IEEE Intl. Conf. on e-Technology, e-Commerce and e-Service. IEEE Computer Society

Brodie M, Bussler C, de Brujin J, Fahringer T, Fensel D, Hepp M, Lausen H, Roman D, Strang T, Werthner H, Zaremba M (2005) Semantically enabled service-oriented architectures: a manifesto and a paradigm shift in computer science. Technical Report TR-2005-12-26. DERI—Digital Enterprise Research Institute

Chesbrough H, Spohrer J (2006) A research manifesto for services science. Commun ACM 49(7):35–40

DERI (2006) D2v1.3. Web service modeling ontology (WSMO). Final draft

Dorn J, Hrastnik P, Rainer A (2007) Web service discovery and composition for virtual enterprises. Int J Web Serv Res 4(1):23–39

eCo Working Group (1999) eCo architecture for electronic commerce Interoperability

Emig C, Weisser J, Abeck S (2006) Development of SOA-based software systems—an evolutionary programming approach. In: Proceedings of the advanced international conference on telecommunications and international conference on internet and web applications and services. IEEE Computer Society

Gardner T (2003) UML modelling of automated business processes with a mapping to BPEL4WS. In: 1st European workshop on object orientation and web services. Springer

Gordijn J, Akkermans H (2001) Designing and evaluating E-business models. IEEE Intell Syst 16(4):11–17

Gordijn J, Akkermans H (2003a) Does e-business modeling really help?. In: HICSS. IEEE Computer Society

Gordijn J, Akkermans JM (2003b) Value-based requirements engineering—exploring innovative e-commerce ideas. Requir Eng J 8(2):114–134

Gordijn J, Osterwalder A, Pigneur Y. (2005) Comparing two business model ontologies for designing e-business models and value constellations. In: Proceedings of the 18th Bled eConference

Gordijn J, Petit M, Wieringa R (2006) Understanding business strategies of networked value constellations using goal and value modeling. In: Proceedings of the 14th IEEE international requirements engineering conference, RE'06

Hammer M, Champy J (2001) Reengineering the corporation: a manifesto for business revolution. HarperBusiness, New York

Hofreiter B, Huemer C (2004) Transforming UMM business collaboration models to BPEL. In: Proceedings of OTM Workshops 2004. Springer LNCS

Hofreiter B, Huemer C, Kim J.-H (2006a) Choreography of ebXML business collaborations. Information Systems and e-Business Management (ISeB)

Hofreiter B, Huemer C, Zapletal M (2006b) A business collaboration registry model on top of ebRIM. In: Proceedings of the IEEE International Conference on e-business engineering. IEEE Computer Society

Ilger M, Zapletal M (2006) An Implementation to transform Business Collaboration Models to executable Process Specifications. In: Proceedings of the Conference on service-oriented electronic commerce at the MKWI 2006. GI LNI

ISO (2004) Open-edi reference model, 2nd edn. ISO/IEC JTC 1/SC30 ISO Standard 14662

Keller G, Nüttgens M, Scheer A.-W (1992) Semantische Prozessmodellierung auf der Grundlage 'Ereignisgesteuerter Prozessketten (EPK)'

Kim H (2002) Conceptual modeling and specification generation for B2B business processes based on ebXML. SIGMOD Rec.

Kotler P, Keller KL (2005) Marketing management. Prentice Hall, Englewood Clis

Kramler G, Kapsammer E, Kappel G, Retschitzegger W (2005) Towards Using UML 2 for modelling web service collaboration protocols. In: Proceedings of the first international Conference on interoperability of enterprise software and applications

Liegl P, Schuster R, Zapletal M (2006) A UML Profile and Add-In for UN/CEFACT's modeling methodology. Dissertation, University of Vienna

Margaria T, Steffen B (2006) Service engineering: linking business and IT. IEEE Computer Society

Masche P, McKee P, Mitchell B (2006) The increasing role of service level agreements in B2B systems. In: Proceedings of the international Conference on web information systems

Mayer RJ, Benjamin PC, Caraway BE, Painter MK (1995a) A framework and a suite of methods for business process reengineering. Business process reengineering: a managerial perspective, pp. 245–290

Mayer RJ, Painter MK, Lingineni M (1995b) Concurrent engineering (IICE): toward a method for business constraint discovery (IDEF9). Knowledge Based Systems, Inc.

McCarthy WE (1982) The REA accounting model: a generalized framework for accounting systems in a shared data environment. The Accounting Review

Mendling J, Hafner M (2005) From inter-organizational workflows to process execution: generating BPEL from WS-CDL. In: Proceedings of OTM 2005 Workshops. Springer LNCS

OASIS (2002) Collaboration-protocol profile and agreement specification. Version 2.0

OASIS (2005a) ebXML registry information model. Version 3.0

OASIS (2005b) ebXML registry services and protocols. Version 3.0

OASIS (2006) OASIS ebXML messaging services. Version 3.0, Public Review Draft 01

OASIS (2007) Web services business process execution language. Version 2.0

Object Management Group (OMG) (2006) Business process modeling notation specification. Version 1.0

Object Management Group (OMG) (2007) Unified modeling language specification. Version 2.1.1

Object Management Group (OMG) (2005) Unified modeling language specification. Version 1.4.2

Osterwalder A, Pigneur Y (2002) An e-business model ontology for modeling e-business. In: Proceedings of the 15th Bled Electronic Commerce Conference

O'Sullivan J, Edmond D, ter Hofstede AHM (2000) Service description: a survey of the general nature of services. report FIT-TR-2003-02

Pateli AG, Giaglis GM (2003) A framework for understanding and analysing e-business models. In: Proceedings of the 16th Electronic Commerce Conference

Porter ME (1979) How Competitive Forces Shape Strategy. Harvard Bus Rev 57(2):137–145

Rainer A (2006) Web service composition using logic programming. Dissertation, Vienna University of Technology, Faculty of Informatics

Rao J, Su X (2004) A survey of automated web service composition methods. In: Proceedings of first international workshop on semantic web services and web process composition. Springer LNCS

RosettaNet (2002) RosettaNet implementation framework: core specification. V02.00.01

Scheer A.-W, Jost W, Hess H (2006) Corporate performance management. ARIS in Practice. Springer, Heidelberg

Sheth A, Verma K, Gomadam K (2006) Semantics to energize the full services spectrum. Commun. ACM

Skene J, Lamanna DD, Emmerich W (2004) SLAng: a language for defining service level agreements. In: Proceedings of the 26th international Conference on software engineering. IEEE Computer Society

Terai K, Izumi N, Yamaguchi T (2003) Coordinating web services based on business models. In: Proceedings of the 5th international conference on electronic commerce. ACM Press

Timmers P (1998) Business models for electronic markets. EM—Electronic Markets

UN/CEFACT TBG14 (2003) UN/CEFACT common business process catalog. Version 0.95

UN/CEFACT TMG (2003a) Core components technical specification. Version 2.01

UN/CEFACT TMG (2003b) UN/CEFACT—ebXML business process specification schema. Version 1.11

UN/CEFACT TMG (2006) UN/CEFACT's Modeling methodology (UMM), UMM Meta Model—foundation module. Technical specification, Version 1.0

W3C (2004) OWL-S: Semantic markup for web services. Member submission

W3C (2005a) Web service semantics—WSDL-S. Member submission

W3C (2005b) Web services choreography description language. Candidate recommendation, Version 1.0

W3C (2007) Service modeling language, Version 1.0. Member submission

Workflow Management Coalition (WfMC) (2005) Process definition interface—XML process definition language. Version 2.0

Zeithaml V, Bitner MJ, Gremler DD (2005) Services marketing. McGraw-Hill/Irwin, NY

Ziemann J, Mendling J (2005) EPC-based modelling of BPEL processes: a pragmatic transformation approach. In: Proceedings of the 7th international conference "Modern Information Technology in the Innovation Processes of the Industrial Enterprises"