A Survey of B2B Methodologies and Technologies: From Business Models towards Deployment Artifacts

Jürgen Dorn[1,2], Christoph Grün[1], Hannes Werthner[1,2], and Marco Zapletal[1]

[1] Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria
{juergen.dorn, christoph.gruen, hannes.werthner, marco.zapletal}@ec.tuwien.ac.at

[2] EC3 - Electronic Commerce Competence Center, Vienna, Austria
{juergen.dorn, hannes.werthner}@ec3.at

Abstract— In recent years business-to-business (B2B) e-commerce has been subject to major rethinking. A paradigm shift from document centric file-based interchange of business information to process-centric and service-based information exchange can be observed. On a business level, a lot of work has been done to capture business models and collaborative business processes of an enterprise. On a technical level, the focus in software development is moving towards service-oriented architectures (SOA). These transitions on both levels promise a market entry at lower costs and an easier adjustment to changing market conditions. Hence, an overwhelming quantity of specifications and approaches emerged in the past targeting the area of B2B - these are partly competing and overlapping. In this paper, we provide a survey of the most promising ones at both levels and classify them using the Open-edi reference model standardized by ISO. Furthermore, we discuss how individual specifications on different levels fit together - starting from business models via business processes to artifacts ready for deployment.

I. Introduction

Traditional approaches in B2B e-commerce had a strong document-centric focus. Business partners were required to agree on exchanged business documents in order to conduct electronic businesses. Syntactic and semantic differences resulted in the need for bilateral agreements on data and protocols between each pair of business partners. Moreover, the implementation and operation of traditional Electronic Data Interchange (EDI) systems - often communicating over special networks - were costly. In consequence, the participation in B2B e-commerce was reserved to larger enterprises that were able to afford the costs.

In order to overcome the shortcomings of the traditional document-centric focus, a lot of research has been undertaken to move towards process-centric B2B approaches. Process-centric approaches capture the flow of business information between business partners, which is communicated 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. Business process models allow smaller companies to participate in established business processes by aligning their internal process to the collaborative process. In addition, business processes might be subject to standardization efforts in the future, which allow a cost-effective implementation of commercial off-the-shelf software (COTS) supporting these standardized processes.

Regarding software development, we observe a major paradigm shift towards service-based communication - known as service-oriented architectures (SOA). The SOA concept allows an easier implementation of communication across heterogeneous platforms and among enterprises based on open and free specifications. Furthermore, SOA utilizes so-called 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. One should note that the meaning of service in the context of SOA differs from the term used in management science. 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 [1] [2] and 'generated' by business processes. In the context of SOA, a service is a unit of executable code representing a certain task. Services can be used to compose more complex services.

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 adopt to changing and newly emerging technologies. The B2B-field offers potential for both research and industrial efforts. Thus, on both levels an overwhelming quantity of specifications and approaches has emerged.

The objective and the major contribution of this paper is to give an overview and a classification of these B2B approaches. We show both, business- and implementation-related specifications and classify them in terms of the Open-edi reference model [3]. We discuss advantages and drawbacks, outline over-

laps and sketch their roots and progressions. Furthermore, we outline the path from a business model down to deployments artifacts for SOA environments. In other words, we discuss the derivation of machine-understandable deployment artifacts on a lower level from models at a higher level capturing the business logic. This is inline with [4] focusing at service driven innovation of businesses, the so-called service sciences.

The remainder of this paper is structured as follows: Section II introduces the Open-edi reference model and the refinements made for our paper. In section III we outline business-related approaches and in section IV we review the most promising SOA related specifications. Section V provides a short look into our future work, where we focus on the (semi-)automation of the flow from business models via business process models to web services, enriched by semantic enhancement. Section VI concludes the paper.

II. REFINING THE OPEN-EDI REFERENCE MODEL

The Open-edi reference model - standardized by the International Organization of Standardization (ISO) - groups EDI related standards into two categories. 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 methodologies. Since the BOV describes a business in a technology independent way, different FSV implementations (i.e., deployments for different platforms) may be derived.
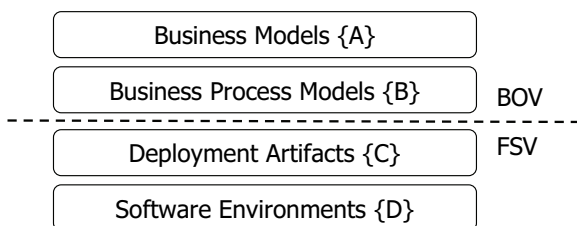

Fig. 1. Classification schema based on refinements of the Open-edi reference model

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 Figure 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 sections III and IV, respectively. The references in braces in Figure 2 are used as pointers to the description of each specification.

III. BUSINESS-RELATED SPECIFICATIONS

On the upper layer, we differentiate between business models and business process models. Business models describe the exchange of values 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 concretize 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* while a business process model defines the *how* [5].

A. Business Models

In the definition of business models, we follow Timmer [6], 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 [7] 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 [8], ranging from establishing taxonomies of business models, to describing elements of business models, and finally to building business models 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 [9] or by building a business object ontology from the e-Business Process Handbook of MIT [10]. 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.

1) e^3-value Model ({1} in Figure 2): In e^3-value [11] [12] [13] a business model is regarded as a value constellation, i.e., a network of enterprises that jointly create and distribute
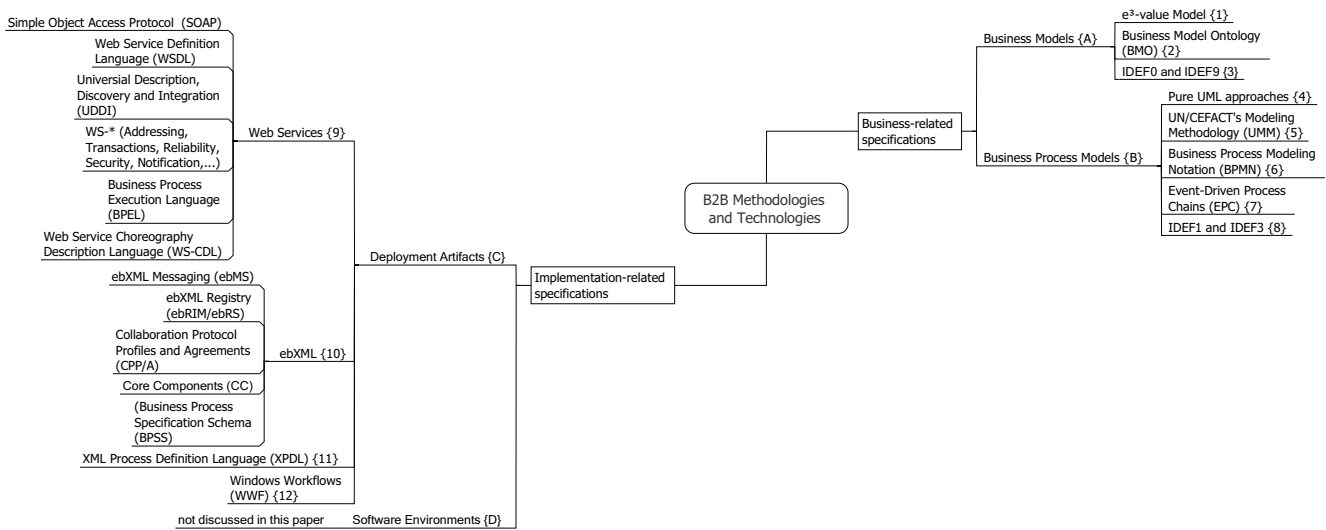
Fig. 2.   Overview of business- and implementation related B2B specifications

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 can 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 [14] 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 the model can be evaluated by means of profit sheets and sensitive analysis.

2) Business Model Ontology (BMO) {2}: In BMO [15], business models are described based on four elements, 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 can make profit. In this way, the element *product innovation* mainly refers to the value the actor offers to a specific target customer segment whereas the element *customer relationship* refers to the different distribution channels to deliver the created value. The element *infrastructure management* outlines how this value is created by regarding the resources of the actor

as well as his partner network. The element *financial aspect* is influenced by all these three elements and determines the actor's profit model.

In [8], the two business models 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 [8] state, the integration of different ontologies seems to be a promising way of combining the advantages of different models.

B. 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 [16]. A business process model is an abstract description of the flow of one or more business processes.

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 [17] to rather petri-net like semantics in UML 2 [18]. Nevertheless, activity diagrams of both major UML versions are adopted for business process modeling.

In [19], Kim suggests an 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 [20], 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 [21]. 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 [22], 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.

2) UN/CEFACT's Modeling Methodology (UMM) {5}: UMM is a UML-based methodology, defined as a UML 1.4.2 profile [23] - 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 [24] 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. 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 [25] and BPEL [26] are proposed. Academic tool implementations exist that generate these types of process specifications based on the proposed transformation rules [27] [28].

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 [29].

Nevertheless, UMM has also shortcomings. Currently, there is a lack of tool support by software vendors. Criticism targets also 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) 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 [30].

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 business process modeling notation, it has a definite business centric approach. According to [31], 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.

4) Event-Driven Process Chains (EPC) {*7*}*:* EPCs are a process-oriented modeling technique proposed by Keller et al. in [32]. 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 [33]. 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 event-driven process chains support the modeling of resources, data objects, organizational units and services. Further, the linkage between processes is supported.

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 use IDEF mainly for process modeling, we discuss 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 [34]. 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 is an object-oriented modeling approach whose advantage should be that it fits into the other parts of the suite. IDEF5 supports the modeling of ontologies. A further extension is IDEF9 [35],

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 shall support business reengineering in enterprises.

IV. 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 consume deployment artifacts in order to execute or participate in a certain process. Due to space limitations, we do not discuss the latter.

A. Deployment Artifacts

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 has 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)

[36] and the Web Service Choreography Description Language (WS-CDL) [37].

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. 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 Web Service Choreography Description Language (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 (e.g., BPEL) might be derived from the global choreography in order to facilitate and verify local implementations. Furthermore, the choreography description can 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. In addition, WS-CDL strives for compatibility with other specifications like BPEL and BPSS. However, no approaches currently exist to derive BPEL descriptions from WS-CDL nor to generate BPSS from WS-CDL and vice versa.

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. ebXML is a distinct process-centric approach influenced by the lessons learned from traditional EDI standards. 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 Business Process Specification Schema (BPSS).

The ebXML messaging [38] 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. Two facets of message reliability are addressed in ebMS. Firstly, ebMS guarantees the successful delivery of a message once and only once, implemented by the concepts of message acknowledgments. Secondly, reliability features maintain the correct message sequence in a collaboration. In addition, ebMS provides integrity, authenticity and authentication by supporting message encryption and digital signatures.

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) [39], defines how actual content is associated with meta data and classifications is . 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) [40]. Internal services a registry must implement are, for example, lifecycle management, versioning, content cataloging and validation and access control mechanisms.

Collaboration protocol profiles (CPP) [41] 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. Business processes and supported roles are specified by referring to business process specifications (i.e., an instance of a BPSS which is discussed later in this section) and to the respective role. 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 timeouts and security requirements. A collaboration protocol agreement (CPA) [41] captures an agreement between two business partners on a certain business process under covenant technical parameters. A CPA represents a kind of intersection between the respective CPP's of both partners.

Core components (CC) [42] 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) [43] 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. BPSS has been closely aligned to UMM by its specification as an executable subset of UMM. It provides a representation for these parts of the UMM that are required to execute a collaborative process (i.e. collaborations, transactions and exchanged documents), but omits parts of the model like captured requirements.

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 core components 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. We accredit this not to the quality of the specifications, but to the missing industry support.

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 [44]. 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 subflow 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.

4) Windows Workflow Foundation {12}: The Windows Workflow Foundation (WWF) is an upcoming technology that will be integrated into the next generation of Microsoft operating systems. It is a declarative approach for defining workflows within or across applications. The WWF framework comprises basically three parts: the workflow engine for executing workflows, services for communicating with workflows and a set of basic workflow activities. Custom and more complex activities might be composed of basic activities in conjunction with custom code.

The WWF provides two workflow styles, sequential workflows and state machine workflows. Sequential workflows define a fixed execution order of activities. Such workflows are mostly applicable to automated processes. State-machine workflows are event driven, which allows the workflow to respond on interactions. This style is rather useful for scenarios in which interactions and decisions by humans are required. However, both styles might also be mixed in the same workflow.

There are three different approaches to specify a workflow. Firstly, one might choose the pure declarative approach using XAML, an XML-based language. Secondly, the workflow might be coded completely in a .NET language. Thirdly, XAML and .NET code might also be mixed. The WWF framework also provides the possibility to generate workflows - using one of three approaches - from another application. This allows the generation of workflows from business process models. More specifically, one can use a collaborative business process model to derive complementary workflows for all participants. The resulting workflows correspond to the abstract processes of each participant. They are expected to interact according to the choreography defined in the business process model. In addition, internal processes can easily be tied to the generated abstract processes in order to support collaborations. In short, the WWF technology provides a promising approach to derive complementary partner specific workflows from collaborative process models.

V. FROM BUSINESS MODELS TOWARDS DEPLOYMENT ARTIFACTS

Flexible service design and implementation requires a (semi-) automated path from business models down to business process models and finally, to deployment artifacts. Hence, these layers must be linked in order to derive artifacts at a lower layer from formal descriptions at a higher layer. In essence, the objective is to abstract technical implementation details from business users. We are aware that this next

abstraction represents a real challenge for information science [45].

Figure 3 expresses this transformation where a collaborative business network manager defines the business perspective that serves as the starting point for generating and executing the respective services and resources.
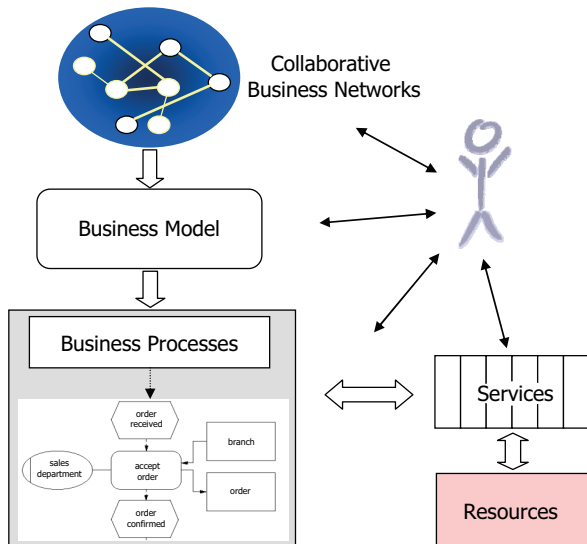

Fig. 3. B2B transformation process: from business models to web services

Partners within a network define their business needs and models in order to describe what 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. This enables potential business partners to find each other and to conduct business according to the respective collaborative processes. One should note that an automatic generation of a process model, describing a flow of activities, from a business model that denotes estimations for needs and wants, is still not 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 for each business partner from a collaborative process model. We outlined approaches for a model-based generation in section III-B in 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. Deployment artifacts are then used within software environments, enabling a configuration at run-time.

The important issue is to obtain a composed and/or choreographed 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 [46]. 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. In principle, such a solution can be deployed as a workflow. A goal could be, for example, that a customer receives his ordered product. Typically, AI-based planners search backward-oriented, thus first looking for an action that has a result matching the goal (e.g., customer has product). Then the preconditions of the action are investigated and a plan is searched for that fulfills the preconditions (often called subgoals of the planning process). 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 service-oriented architectures 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 [47], 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. A prototype implementation was applied in several concourses of web service composition contests [48] with very good performance. However, the solution is restricted to small to medium size problems since all solutions are considered. Another problem in web service composition is the nonexistence of semantic annotations of web services (see later). In the existing solutions a restricted vocabulary of input and output arguments is used that helps to match goals and existing web services. Moreover, predefined web service choreographies must be considered by the planner.

The latter approach also represents a promising way to (semi-)automatically derive deployment artifacts from business objectives. We believe that the level of automation in

the transformation process (see Figure 3) can be enhanced by adding semantics to the individual layers, using service-oriented computing as the underlying computing paradigm. A SOA is essentially a collection of services, enabling distributed computing by abstracting from software and by regarding all resources as services. Such collections can be large - a service-oriented world will likely have millions of services. Computation will involve searching for services based on functional and non-functional requirements and interoperating 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.

A lack of formal semantics prevents automated partner discovery, systems integration, and process monitoring. On top of that, the insufficient level of abstraction limits the reuse of existing process models in new contexts. As a consequence, semantic web service frameworks, namely the Web Service Modeling Ontology (WSMO) [49], OWL-S [50], and WSDL-S [51] are gaining ground. These technologies can be used as the starting point to raise the degree of automation of the transformation process.

VI. CONCLUSION

In this paper, we presented an overview of approaches, methodologies, specifications and technologies in the area of B2B e-commerce. We classified them based on the Open-edi reference model, which we refined into the layers: *business models*, *business processes*, *deployment artifacts*, and *software environments*. When creating new 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. In our opinion, little research has tempted to integrate the different layers. Moreover, scientists from economics and computer science have different points of view and definitions regarding services. The goals of this paper 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. We highlighted the possible paths from the description of business logic in form of business models to the derivation of software artifacts. Doing so, we showed that the gap between the business and technical perspective of B2B e-commerce is becoming smaller, but still exists. We foresee that this gap might be closed by using semantics, currently a hot topic in research.

REFERENCES

[1] V. Zeithaml, M. J. Bitner, and D. D. Gremler, *Services Marketing*. McGraw-Hill/Irwin, New York, NY, May 2005.

[2] P. Kotler and K. L. Keller, *Marketing Management*. Prentice Hall, Englewood Clis, NJ, Mar. 2005.

[3] *Open-edi Reference Model*, ISO, 2004, ISO/IEC JTC 1/SC30 ISO Standard 14662, Second Edition.

[4] H. Chesbrough and J. Spohrer, "A Research Manifesto for Services Science," *Commun. ACM*, vol. 49, no. 7, pp. 35–40, 2006.

[5] M. Bergholtz, P. Jayaweera, P. Johannesson, and P. Wohed, "Process Models and Business Models - A Unified Framework." in *Proceedings of ER (Workshops) 2002*. Springer, Lecture Notes in Computer Science, 2002, pp. 364–377.

[6] P. Timmer, "Business Models for Electronic Markets," *EM - Electronic Markets*, vol. 8, no. 2, July 1998.

[7] G. G. A. Pateli, "A Framework for Understanding and Analysing e-Business Models," in *Proceedings of the 16th Electronic Commerce Conference - eTransformation*, June 2003.

[8] J. Gordijn, A. Osterwalder, and Y. Pigneur, "Comparing Two Business Model On tologies for Designing e-Business Models and Value Constellations," in *Proceedings of the 18th Bled eConference - eIntegration in*, June 2005.

[9] W. E. McCarthy, "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment," *The Accounting Review*, 1982.

[10] K. Terai, N. Izumi, and T. Yamaguchi, "Coordinating web services based on business models," in *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*. New York, NY, USA: ACM Press, 2003, pp. 473–478.

[11] J. Gordijn and H. Akkermans, "Designing and evaluating e-business models," *IEEE Intelligent Systems*, vol. 16, no. 4, pp. 11–17, 2001.

[12] J. Gordijn and J. M. Akkermans, "Value-based requirements engineering: exploring innovative e-commerce ideas." *Requir. Eng.*, vol. 8, no. 2, pp. 114–134, 2003.

[13] J. Gordijn and H. Akkermans, "Does e-business modeling really help?" in *HICSS*, 2003, p. 185.

[14] J. Gordijn, M. Petit, and R. Wieringa, "Understanding business strategies of networked value constellations using goal and value modeling," in *accepted at the 14th IEEE International Requirements Engineering Conference, RE'06*, 2006.

[15] A. Osterwalder and Y. Pigneur, "An e-Business Model Ontology for Modeling e-Business," in *Proceedings of the 15th Bled Electronic Commerce Conference - e-Reality*, June 2002.

[16] M. Hammer and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*. New York (NY): HarperBusiness, 2001.

[17] *Unified Modeling Language Specification*, Object Management Group (OMG), Apr. 2005, version 1.4.2. [Online]. Available: http://www.omg.org/docs/formal/05-04-01.pdf

[18] *Unified Modeling Language Specification*, Object Management Group (OMG), July 2007, version 2.0. [Online]. Available: http://www.omg.org/docs/formal/05-07-04.pdf

[19] H. Kim, "Conceptual modeling and specification generation for b2b business processes based on ebxml," *SIGMOD Rec.*, vol. 31, no. 1, pp. 37–42, 2002.

[20] G. Kramler, E. Kapsammer, G. Kappel, and W. Retschitzegger, "Towards Using UML 2 for Modelling Web Service Collaboration Protocols," in *Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA'05)*, Feb. 2005.

[21] *eCo Architecture for Electronic Commerce Interoperability*, eCo Working Group, 1999.

[22] T. Gardner, "UML Modelling of Automated Business Processes with a Mapping to BPEL4WS," in *1st European Workshop on Object Orientation and Web Services (EOOWS'03)*. Springer, 2003.

[23] *UN/CEFACT's Modeling Methodology (UMM), UMM Meta Model - Foundation Module*, UN/CEFACT Techniques and Methodologies Group, Mar. 2006, Candidate for 1.0, Second Working Draft. [Online]. Available: http://www.unece.org/cefact/umm/UMM_Foundation_Module.pdf

[24] *RosettaNet Implementation Framework: Core Specification*, RosettaNet, Dec. 2002, v02.00.01. [Online]. Available: http://www.rosettanet.org/rnif

[25] B. Hofreiter, C. Huemer, and J.-H. Kim, "Choreography of ebXML business collaborations," *Information Systems and e-Business Management (ISeB)*, June 2006.

[26] B. Hofreiter and C. Huemer, "Transforming UMM Business Collaboration Models to BPEL," in *Proceedings of OTM Workshops 2004*, vol. 3292. Springer LNCS, 2004, pp. 507–519.

[27] M. Ilger and M. Zapletal, "An Implementation to transform Business Collaboration Models to executable Process Specifications," in *Proceedings of the conference on Service-Oriented Electronic Commerce at*

IEEE
COMPUTER
SOCIETY

the Multikonferenz Wirtschaftsinformatik 2006, Passau Germany, ser. GI-Edition - Lecture Notes in Informatics (LNI). Gesellschaft fuer Informatik (GI), Feb. 2006, pp. 9–23.

[28] P. Liegl, R. Schuster, and M. Zapletal, "A UML Profile and Add-In for UN/CEFACT's Modeling Methodology," Master's thesis, University of Vienna, February 2006. [Online]. Available: http://www.ifs.univie.ac.at/ummaddin/downloads/thesis/AUMLProfileandAddInforUMM_20060214.pdf

[29] B. Hofreiter, C. Huemer, and M. Zapletal, "Registering UMM Business Collaboration Models in an ebXML Registry," in *Proceedings of the IEEE Conference on E-Commerce Technology (CEC '06)*. IEEE Computer Society, June 2006. [Online]. Available: http://www.cs.univie.ac.at/upload/shared/mis/publications/IEEE-CEC06-final.pdf

[30] *Business Process Modeling Notation Specification*, Object Management Group (OMG), Feb. 2006, version 1.0. [Online]. Available: http://www.omg.org/docs/dtc/06-02-01.pdf

[31] C. Emig, J. Weisser, and S. Abeck, "Development of SOA-Based Software Systems - an Evolutionary Programming Approach," in *AICT-ICIW '06: Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services*. IEEE Computer Society, 2006, p. 182.

[32] G. Keller, M. Nüttgens, and A.-W. Scheer, "Semantische Prozessmodellierung auf der Grundlage 'Ereignisgesteuerter Prozessketten (EPK)'," Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes Heft 89.

[33] A.-W. Scheer, W. Jost, and H. Heß, *Corporate Performance Management. ARIS in Practice*. Springer, Berlin, 2006.

[34] R. J. Mayer, P. C. Benjamin, B. E. Caraway, and M. K. Painter, "A Framework and a Suite of Methods for Business Process Reengineering," *Business Process Reengineering: A Managerial Perspective*, pp. 245–290, 1995.

[35] R. J. Mayer, M. K. Painter, and M. Lingineni, "Concurrent Engineering (IICE): Toward a Method for Business Constraint Discovery (IDEF9)," Knowledge Based Systems, Inc.

[36] *Business Process Execution Language for Web Services*, BEA, IBM, Microsoft, SAP AG and Siebel Systems, May 2003, Version 1.1. [Online]. Available: ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf

[37] *Web Services Choreography Description Language*, World Wide Web Consortium (W3C), Nov. 2005, Version 1.0. [Online]. Available: http://www.w3.org/TR/ws-cdl-10/

[38] *OASIS ebXML Messaging Services*, OASIS, May 2006, Version 3.0, Public Review Draft 01.

[39] *ebXML Registry Information Model*, OASIS, May 2005, Version 3.0.

[40] *ebXML Registry Services and Protocols*, OASIS, May 2005, Version 3.0.

[41] *Collaboration-Protocol Profile and Agreement Specification*, UN/CEFACT and OASIS, May 2001, Version 1.0.

[42] *Core Components Technical Specification - Part 8 of the ebXML Framework*, UN/CEFACT Techniques and Methodologies Group, Nov. 2003, version 2.01.

[43] *UN/CEFACT - ebXML Business Process Specification Schema*, UN/CEFACT, Nov. 2003, Version 1.11.

[44] *Workflow Process Definition Interface - XML Process Definition Language*, Workflow Management Coalition, 2002, Version 1.0.

[45] M. Brodie, C. Bussler, J. de Brujin, T. Fahringer, D. Fensel, M. Hepp, H. Lausen, D. Roman, T. Strang, H. Werthner, and M. Zaremba, "Semantically Enabled Service-Oriented Architectures: A Manifesto and a Paradigm Shift in Computer Science," DERI - Digital Enterprise Research Institute, Tech. Rep. TR-2005-12-26, Dec. 2005. [Online]. Available: http://www.deri.at/fileadmin/documents/DERI-TR-2005-12-26.pdf

[46] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," in *In Proceedings of First International Workshop on Semantic Web Services and Web Process Composition*, ser. Lecture Notes in Computer Science, vol. 3387, 2004, pp. 43–54.

[47] A. Rainer, "Web Service Composition using Logic Programming," Ph.D. dissertation, Vienna University of Technology, Faculty of Informatics, 2006.

[48] M. B. Blake, K. C. Tsui, and A. Wombacher, "The EEE-05 Challenge: A New Web Service Discovery and Composition Competition," in *IEEE Intl. Confernce on e-Technology, e-Commerce and e-Service*. IEEE Computer Society, 2005, pp. 780–781.

[49] *D2v1.2. Web Service Modeling Ontology (WSMO)*, Web Service Modeling Ontology Working Group, Apr. 2005, final Draft 13 April 2005.

[50] DAML Services, "Owl-s website," 2005, http://www.daml.org/services/owl-s/. [Online]. Available: http://www.daml.org/services/owl-s/

[51] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.-T. Schmidt, A. Sheth, and K. Verma, "Web Service Semantics - WSDL-S," Nov. 2005, w3C Member Submission.