UN/CEFACT's Business Collaboration Framework - Motivation and Basic Concepts

Birgit Hofreiter[1], Christian Huemer[1,2] and Klaus-Dieter Naujok[3]

[1] Department of Computer Science and Business Informatics, University of Vienna,
Liebiggasse 4, 1010 Vienna, Austria
{birgit.hofreiter,christian.huemer}@univie.ac.at

[2] Department of Information Systems, University Duisburg-Essen
Universitätsstrasse 9, 45141 Essen, Germany
huemer@wi-inf.uni-essen.de

[3]Global e-Business Advisory Council,
4410 Deermeadow Way, Antioch, CA 94531, USA
Klaus@ge-bac.com

Abstract. Although standards for B2B e-Commerce exist, business partners still need additional bilateral agreements and coding partner-specific adaptations of their applications. Most research is conducted in the area of IT-solutions to interconnect the systems. Less activities are directed towards a shared business logic that allows business servers to collaborate. UN/CEFACT's business collaboration framework (BCF), which is presented in this paper, addresses this point. Instead of connecting two independent systems, it considers a B2B relationship to be a single system with interface on each partner's end. The goal of this paper is to motivate the need for the BCF and to give a basic introduction into the methodology used in BCF.

1 Motivation

The idea to eliminate paper documents for exchanging business data by connecting computer systems of business partners was already created during the Berlin Airlift [Sc88] and became reality in the 1960ies. At this time it was not called Business-to-Business e-Commerce (B2B). It became famous by the term Electronic Data Interchange (EDI) [HF89]. EDI was successfully implemented by the Fortune 1000 companies, but failed in acceptance by small and medium enterprises (SMEs). EDI is said to be too complex and SMEs cannot afford the high start up costs.The success of the Web resulted in a trend towards mark-up languages. Accordingly, XML became the language of choice for data transfer on the Internet. Admittedly, the simple syntax of XML raised interest in implementing B2B solutions. However, we are still missing the widespread success of B2B for SMEs.

This leads to the question, what do the SMEs need to participate in B2B. As a first step, it is necessary to analyze the characteristics of these enterprises. Usually, enterprises are classified into the categories large enterprises, SMEs and micro enterprises according to their number of employees, their turnover and their balance sheet total [Eu02]. However this classification does not seem to be appropriate to classify enterprises according to their B2B capabilities. However, we feel that there exists a correlation between the size of

an enterprise and its enterprise software. Large enterprises either implement their own software or they customize ERP-software to their own needs. SMEs usually buy a software package that fulfils certain business functions, install it and run it. Micro enterprises do not use software packages at all. Since B2B solutions are about inter-computer communications [Fe00], micro enterprises are out of focus.

Today, B2B is commonly based on an interchange file in an agreed format that is imported into (and exported from) the enterprise software. This agreed format usually follows a standard document type. Nevertheless, an additional agreement between the business partners is necessary to customize the standard document type to the partner-specific requirements (c.f. Section 2). As a consequence, business partners have to customize their import/export functions to the partner-specific agreement. Having said that SME install and run their software, they are not able–despite of existing tool support–to implement the import/export function. Only large companies are able to perform this task and, thus, they are the ones that profit from B2B. Therefore, SMEs need software packages that provide both the business functions needed (e.g. purchase order handling) and the B2B functionality (e.g. ordering goods from a business partner). If both functionalities go hand in hand together in low cost commercial off-the-shelf software (COTS), we will see B2B e-commerce to take off.

To enable COTS software production by agreeing on a single data requirement for a particular document type is unrealistic. This will never happen. The Techniques and Methodologies Group (TMG) of the United Nation's Centre for Trade Facilitation and Electronic Business (UN/CEFACT) conducted research to look for alternatives. In TMG we suggested to develop a well-defined business process for each particular business goal, such as "Catalog Ordering", that contains all the possible activities that could be part of that goal. Since these business processes are collaborative in nature, they are called business collaborations. There might be many ways of executing the business collaboration, but each one well defined. Depending on their internal processes, one trading partner may be able to execute all alternatives, where another may only execute a certain number of them. For two trading partners to engage in the same business collaboration, they must both be able to execute at least one alternative in common. In regard to the SMEs, it is envisioned that the software providers will create applications that implement business collaborations with their most popular execution scenarios. This approach requires unambiguous business collaboration models with respect to their choreography and involved document structures. For this purpose, we have developed within TMG UN/CEFACT's Modeling Methodology (UMM) [HH03, UT03]. The UMM builds the heart of the Business Collaboration Framework described in this paper.

The remainder of this paper is structured as follows: Section 2 details the relationship of UMM and other related initiatives in the area of B2B. In Section 3 we present the limitations of current approaches in B2B. We demonstrate the problems of document standards due to overloaded document types. Since Web Services for B2B usually use these document types as input and output technology, we feel that despite of all innovative aspects they are not an enabler to COTS software development. In Section 4 we present the concepts of the BCF and the basic architecture that will enable COTS software solutions. The steps of UMM defining business collaborations and a simple example are provided in Section 5. A short summary and future work items towards COTS B2B software complete the paper in Section 6.

2 UN/CEFACT's BCF and related initiatives

The United Nations's Centre for Trade Facilitation and Electronic Business (UN/CEFACT), previously also called UN/ECE WP.4, is a long-existing B2B standards body. It became famous by developing and maintaining the UN/EDIFACT standards. Since 1995 its techniques and methodologies group (TMG), previously also called TMWG and AC.1, has been responsible to develop their next-generation of EDI. The development of UN/CEFACT's Modeling Methodology (UMM) has always been part of this work. UMM is based on the Open-edi reference model, which became an ISO standard in 1997 (IS 14662). Open-edi distinguishes between a Business Operational View (BOV) and a Functional Service View (FSV) [ISO95]. The BOV is a perspective of business transactions limited to those aspects regarding the making of business decisions and commitments among organizations. The FSV is a perspective of business transactions limited to those information technology interoperability aspects of IT systems that are needed to support the execution of Open-edi transactions. UMM defines a UML Profile [BJR98] for modeling BOV standards in order to capture the business knowledge of business collaborations.

Early in 1999 members of the UN/CEFACT plenary and UN/EDIFACT users asked for an XML solution. The TMG report on this subject rejected the idea of creating 'Yet Another XML Solution' by converting UN/EDIFACT to XML. Instead the recommendation was to built up on UMM by using business process modeling to create Open-edi BOV standards and by using XML as key concept in the Open-edi FSV layer. Furthermore, TMG recommended looking for partnership from the XML industry. UN/CEFACT found OASIS as a partner and started ebXML in November 1999 [HHK02]. Although UMM was the driving edge for UN/CEFACT to enter ebXML, UMM did not become a mandatory part of ebXML. The ebXML architecture specification mentions that, *if implementers and users select to model Business Processes and Information, then they shall use the UMM* [EN01]. Nevertheless a lot of the semantics captured in UMM are also found in ebXML business process specification schema (BPSS), which is an XML schema to describe the choreography of a business collaboration. BPSS is not a modelling methodology, it just enables to write a software module on each business partners side to keep track of the business collaboration.

Many companies and organizations have contributed to the development of UMM. Important industry user groups as EAN-UCC, SWIFT, Telemanagement Forum aligned their methodologies with UMM. The most important contribution was made by the company EDIFECS during the ebXML initiative. EDIFECS developed the business collaboration framework that is used by RosettaNet to develop their partner interface processes (PIPs). In 2000 EDIFECS transferred the copyright of its methodology to UN/CEFACT. The key developers joined us in the TMG to merge the RosettaNet BCF with UMM version 9. Inasmuch, UMM version 10 and higher are not only UN/CEFACT's methodology but also successors of the methodology used in RosettaNet's BCF.

UMM will be the underlying methodology for all UN/CEFACT's B2B solutions to be created in the future. UN/CEFACT restructured itself in 2002 to reflect the changing needs. The International Trade and Business Processes Group (TBG) or better its industry sub-groups are responsible for creating UMM-compliant models. Both UMM and these compliant models build UN/CEFACT's Business Collaboration Framework (BCF). The BCF provides insurance against obsolescence by allowing recasting of the business sce-

narios into new information exchange technologies. The Applied Technology Group (ATG) is responsible for the recast. Currently, the models are mapped to UN/EDIFACT and an XML document standard.

3 Obstacles in B2B Technologies

3.1 Overloaded Document Types

In this Section we demonstrate that the most significant problem of traditional EDI standards were the overwhelming choice of data elements in a document type. However, this problem is independent of the transfer syntax. Instead of learning from the history, XML document standards (see [Li00]) follow the same path. We have selected UN/EDIFACT as representative of traditional EDI standards and xCBL as representative of XML-based vocabularies to demonstrate the common shortcomings [Hu01].

To illustrate heterogeneous interpretations we use the example of specifying parties in a purchase order document. For the purpose of a syntax-neutral analysis, Figure 1 depicts a UML class diagram representing the structure of parties in a UN/EDIFACT purchase order message. Among others the following problems are encountered:

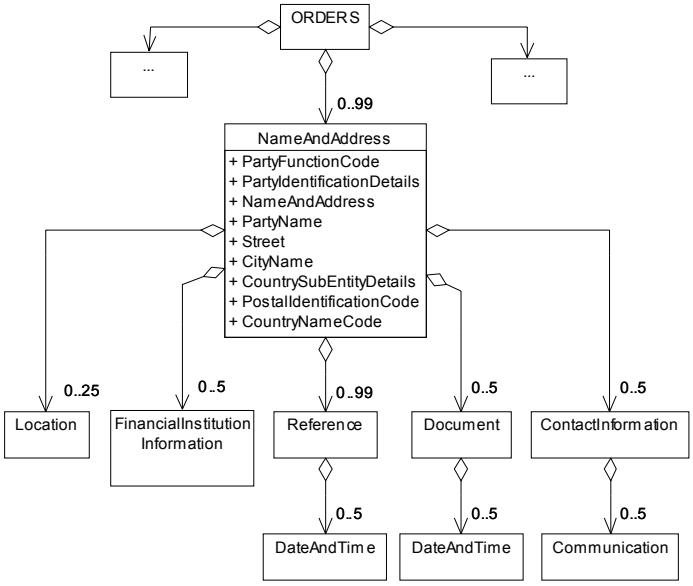

Fig. 1. UML Class Diagram for Parties in anUN/EDIFACT Purchase Order Message Type

- Up to 99 parties might appear in a purchase order. It is not documented which types of party are meaningful to be mentioned in a purchase order. The standard allows to express any type of party listed in the code list for party function code. However, it

does not seem to be useful to specify e.g. a social securities collector's office in a purchase order. Some party types are useful in a purchase order, e.g. buyer's agent, but might not be processable for a particular organization. An agreement has to specify all party types useful in a specific partnership.

- There are multiple ways to identify a party — by a unique id (e.g. customer number) in the party identification details, by unstructured name and address lines, by structuring party name, street, city name etc. The segment/class allows all three choices, but only one must be used. Furthermore, the choice selected might be different for different types of parties. A partner-specific agreement has to fix the details.

- The assignment of multiple addresses to a party is difficult, since both name and address appear in the same segment/class. Thus, it is not possible to assign multiple addresses to a buyer. It is necessary to specify another 'virtual' party for each address. Another choice is the use of the location segment/class, since up to 25 locations might be specified for each party. However, this segment/class is restricted to locations listed in a code list like the UN LOCODE (which covers e.g. ports, airports, etc.).

- Further information assigned to a party includes financial institutions, references, documents, and contacts . A partner agreement has to identify which of them are meaningful for which type of party. Moreover, the problems due to ambiguity hold over the details of these segments/classes. E.g., there exist many types of contacts and each one might be contacted by a different communication channel.

It follows that sending a UN/EDIFACT document conform to the standard does not mean that the receiver of this message supporting UN/EDIFACT is able to process it. Consequently, a message must be in conformance to a partner-specific agreement which defines additional access rules to ensure interoperability.

Figure 2 uses UML to depict the structure of parties in an xCBL purchase order document type. The comparison with the examples mentioned above leads to the following results:

- xCBL explicitly defines the following party types for a purchase order: buyer party, supplier party, ship to party, and bill to party. According to the standard, the buyer and the seller must be included in the document - there is no other choice even if there is no business need to specify them. The other ones are optional. If any further parties are required, they must be specified using the element/class list of party coded. Since the concept is based on a party role code, which is similar to the party function code in UN/EDIFACT, the same problems are encountered in xCBL.

- In xCBL also exist multiple ways to identify a party. The options include the unique identifier and the identification by specifying party name and address in a structured way. There does not exist the choice of unstructured address lines. With respect to the unique identification there are again two options, either the party id attribute or in case of multiple identifiers the substructure of the element/class list of identifier. Again - like in UN/EDIFACT - a partner specific agreement has to fix the identification details for each party type.

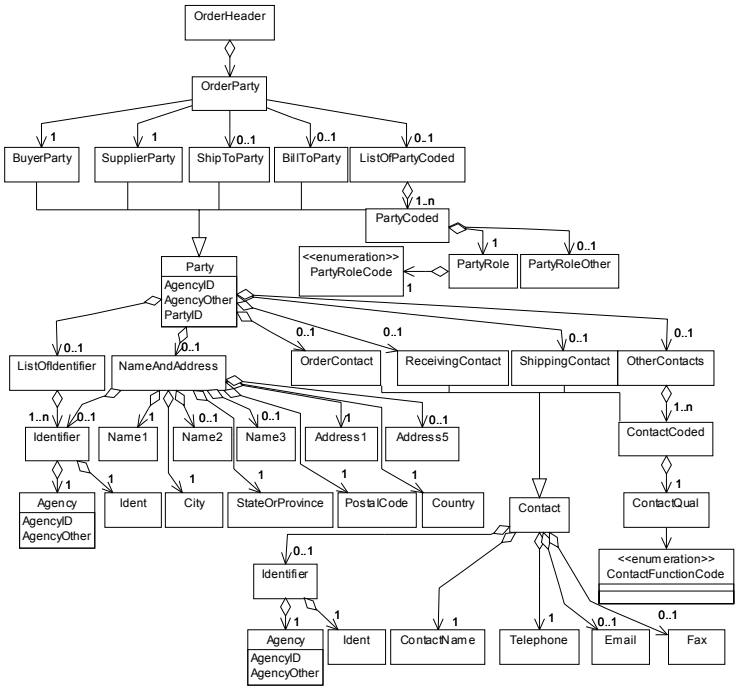

Fig. 2. UML Class Diagram for Parties in an xCBL Purchase Order Document Type

- Assigning multiple addresses to a party also requires the specification of another virtual party for each address. This becomes obvious by the explicitly mentioned party types ship to party and bill to party. Coded locations do not appear in xCBL orders. Anyway, the encountered problems are similar.

- xCBL does not include structures for financial institution information, references and documents. It does include the following optional types of contact: order contact, receiving contact and shipping contact. All other contact types are specified by coded contact functions. The possible communication channels for each contact are telephone, e-mail and fax. Nevertheless, an agreement must define which types of contact are useful for which type of party, not to mention the use of different communication channels for each type of contact.

The comparison demonstrates that—independent of the document standard—business partners have to agree on the data element types to be used in a partnership before they can start doing business electronically. Usually this agreement uses only about 3% of the elements of the standard document types. For current B2B partnerships document instances must not only be compliant to the document type, but also to the partner-specific constraints. If each business partnership requires a different agreement, this leads to a proliferation of variations ($\frac{n^2-n}{2}$) and implementations of the same business document type.

3.2 Web Services

Although XML does not solve the B2B problem, it is undisputable that XML represents the state of the art in B2B. XML and e-Business is considered synonymous with Web Services. Web Services are defined as a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web Service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols [WW02]

Fig. 3 depicts the basic idea of a Web Service Architecture (c.f. [FF03]). A service provider has implemented services for others to use. A so-called WSDL document describes the details of this service in XML, i.e. the operations of the service and the input and output messages for each operation. Furthermore the binding description defines how to send messages on the wire where the service is located. The service description is then published in a registry, such as UDDI, in order that clients can find the service based on a variety of search criteria. Once the requestor finds a suitable service, it uses the service description to develop/configure a client to interact with the service..

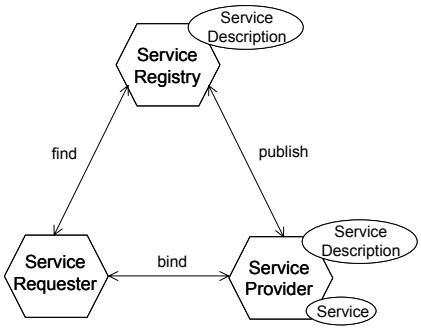

Fig. 3. Web Service Architecture

Today web services are the most promising technology to transform e-Business into a plug-and-play solution. The reason for the increased interest in Web Services is the promise of interoperability, in the same way that Web pages can be accessed from anywhere on the Internet. The promise is simplifying how enterprises will interconnect and, thus, offering a cheap solution to all types of enterprises. The challenging aspect is the necessary effort to develop/configure the client. Since all the service descriptions are XML-based, an automatic configuration is envisioned. As long as Web Services use today's documents standards as input/output to their operations, partner-specific agreements followed by an enormous efforts to configure the client are hurdles on the way to reach real interoperability. B2B requires interoperability, not only at the messaging and transport layer, but also at the business (application) layer. There are many efforts in standardizing the infrastructure for Web Services. Less activities focus on interoperability at the business layer. This aspect is addressed by UN/CEFACT's Business Collaboration Framework (BCF) presented in the following sections.

4 An oo-based Business Collaboration Architecture

Over the last two decades two main contributions have significantly influenced the IT industry: client-server computing and object-oriented computing. In this section we evaluate how these two concepts effect B2B computing.

The Web is based on client-server computing. A web client sends an http request to the web server that processes the request and returns an HTML document in the http response. We do not want to go into the details of refinements/alternatives based on applets, XML, etc. For our analysis, the way that server-side web applications are built is more important. A multi-tier architecture as depicted in Fig. 4 became the preferred architecture. The messaging layer realizes the http-based communication between the server and the client, i.e. a web server. In the presentation layer the user interface for a web application is developed. The most important layer is the business layer that realizes the business logic provided by the web application. It processes the enterprise-critical data. The business logic is usually implemented by business objects that access the persistence layer. The persistence layer is represented by existing databases, ERP systems or any legacy applications in place.

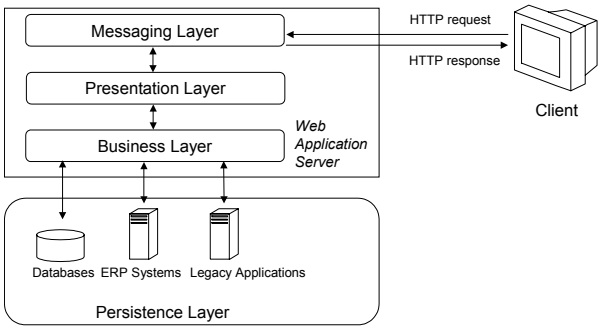

Fig. 4. Web Application Server Architecture

The architecture described above is based on a business collaboration where one business partner uses a web browser to access the web application of the other business partner. In this case the business partner on the server side dictates the business logic of the business collaboration. In a B2B environment usually the applications of both partners must be integrated into the collaboration (see Fig. 5). For a single interaction between the business partners, one side is still the server and the other one the client. Therefore, Web Services represent a client-server technology to implement this interaction. Since the client must process the data as well, the input and output is based on XML instead of HTML. Hence, we use the term document layer instead of presentation layer.

A business collaboration requires multiple interactions between the business partners. It is necessary to identify all interactions of the same business collaboration. This is realized by specific entries in a SOAP envelope (c.f. [PN03]) used as messaging protocol on top of http, smtp, etc. Usually, a business partner represents the server in some interactions, but the client in other interactions. This means that a business collaboration is conceptually a server-server communication, that is implemented by a choreography of client-server interactions. As a consequence, the business logic does not sit on a single side of

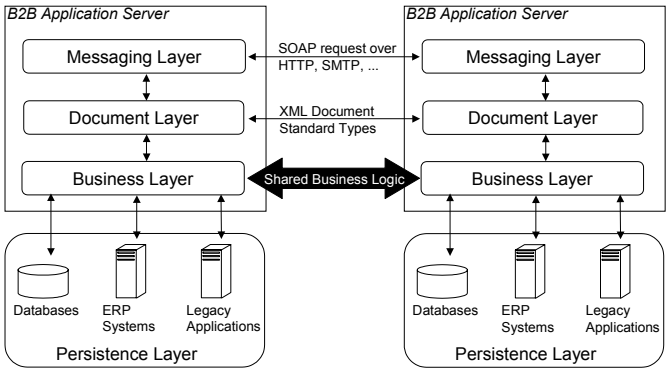

Fig. 5. B2B Application Server Architecture

the collaboration. It is necessary that business partners share the business logic of the business collaboration. Both the choreography of interactions and the input/output documents must be unambiguous.

Today, too much focus is spent on the document layer instead of the business layer. Document standards are built by reverse engineering data base structures of standards participants. This leads to the overloaded document types represented in a specific document standards language. Similarly, to the client-server based approach the business layer is the most important one in the B2B environment. Once a common business logic is agreed, it is comparatively simple to switch from one document standard to the other as from one user interface to the other. Starting-off with data base structures or paper-based documents in order to define the business logic is not an appropriate approach. It is more appropriate to design a collaboration as a well-defined choreography of business state changes. The information needed to change from one business state to the other will be communicated in the interchange. That is what is implemented in the client-server based browser applications. E.g. ordering items from an offered item list, requires to make reference to the line numbers, not to send an overloaded order document.

UMM provides a modeling methodology to define the business logic. In order to develop a shared business logic it is necessary to regard the business collaboration as a single inter-organizational system (see Fig. 6).Today, the object-oriented approach is the preferred choice for designing and implementing in-house systems. An application is built by collaborating objects. When designing the applications, different designers will not develop their classes in isolation. Otherwise the objects would not collaborate. Designers clearly define how one object calls the operation of another. A similar approach is necessary for a B2B system. A B2B system consists of the private areas on each partner's side and a collaborative space in-between. The B2B functionality of each participating partner must be realized by a partner interface class sitting at the border of the private and the collaborative space. A business collaboration must be built by collaborating partner interface classes. When designing the business collaboration, designers of different partner types must not develop their partner interface classes in isolation. Otherwise the interface classes will not collaborate. B2B standard specifications must unambiguously define the collaborative space of a business collaboration, or in other words define the collaboration of partner interface classes. They must not consider the private space.

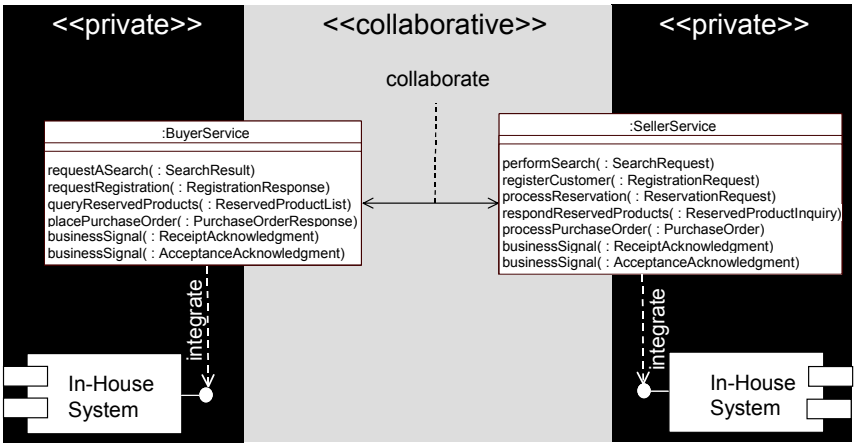

Fig. 6. An object-oriented B2B system

It is expected that software vendors will use the unambiguous partner interface class definitions to integrate them into their products supporting a corresponding role in a business collaboration.

5 The Business Collaboration Framework

The primary goal of the BCF is to capture the business knowledge that enables the development of low cost software components to help the SMEs engage in e-Business practices. A commercial trading agreement is modeled as a business collaboration model according to the UMM. The BCF comprises a set of architectures, patterns and business semantics defined in accordance with certain business reference models and ontologies. The framework consists of 4 views, corresponding patterns, as well as a well-formed meta-model which defines the syntax and semantics for each view. Uniformity of notation and precision of semantics provide concise and unambiguous business process definitions. In this section we will present the 4 views.

5.1 Business Domain View

The business domain view (BDV) is used to gather existing knowledge. It identifies the business processes in the domain of the business problems that are important to stakeholders. It is important at this stage that business processes are not constructed, but discovered. Stakeholders might describe intra-organizational as well as inter-organizational business processes. All of this takes place in the language of the business environment experts and stakeholders. Hence any ontology is appropriate and no patterns apply for this step. In our simple example, which is ordering from a punch out catalog, potential sellers and buyers will describe their needs. The business domain view is modeled in a use case diagram. The BCF provides worksheet templates to describe the discovered processes that are represented as use cases. We do not depict the use case diagram due to space limitations.

5.2 Business Requirements View

The goal of the business requirements view (BRV) is to identify possible business collaborations in the considered domain and to detail the requirements of these collaborations. Business collaborations span over multiple business processes discovered in the previous workflow. Thus, a use case for a business collaboration must consider the views of different stakeholders. The description of the use case must present an harmonized view on the business collaboration being developed.

The business collaboration is an elaboration of business scenarios, resources, business events, and constraints. It defines the semantics that enterprises use to describe their collaborative units of work. These activities represent the processes and resources used to achieve certain definable goals or objectives, the economics of a system. Resources, events (indication of a process result) and agents (participants) are the key elements of an economic ontology known as the REA enterprise ontology [Mc03]. TMG is currently

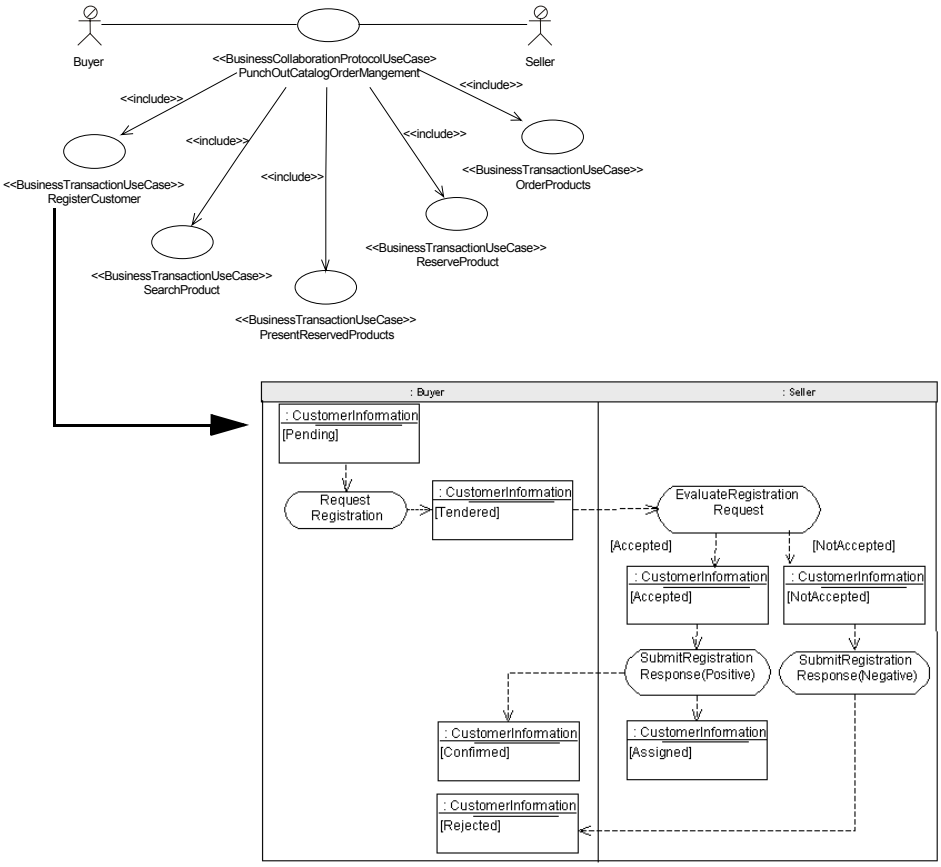

Fig. 7. Business Requirements View for the Catalog Order Example and Business States for the Register Customer Transaction

working on business collaboration patters for e.g. negotiation, order-fulfillment-settlement, long term contract with periodic releases, to name just a few.

Fig. 7 shows typical artifacts of the BRV for our reference example. The use case diagram on the left shows the business collaboration use case for the punch out catalog order management. Again a worksheet helps to describe the use case. According to the description the use case involves the registration of the customer, the search for products, the reservation of products, the presentation of reserved products, and the ordering of products. These represent collaborative use cases that are included in the order management and are themselves described by a worksheet. Since it turns out that they are realized by a single or a pair of business information exchange, they are stereotyped as business transaction use cases.

As we mentioned in the previous section, a business collaboration is best designed by a choreography of business state changes. Thus, it is important to analyze the effects of activities on the business state of the collaboration or, better, on the business states of business entities, which have a life-cycle during the collaboration. The right part of Fig. 7 depicts the business state changes for the register customer transaction. It defines preconditions, post-conditions and inter-mediate states of the customer information during the transaction.

5.3 Business Transaction View

The Business Transaction View (BTV) represents the view of the business process analyst. In the first step the business collaboration is modeled according to the requirements identified in the BRV. An activity diagram called business collaboration, shows the choreography amongst the business transactions. The transitions in the business collaboration diagram depend always on the business states of the collaboration or, better its business entities. Fig. 8 shows an extract of the business collaboration protocol of the reference example.

The next step in the BTV is to detail each business transaction by a separate activity graph. This defines the choreography within a business transaction. The reference ontology for this step comes from "The Commercial use of Electronic Data Interchange, Section of Business Law American Bar Association, A report and model trading partner agreement" [UCC92] and "Part 2 Uniform Rules of Conduct for Interchange of Trade Data by Teletransmission (UNCID), Chapter 2 - Text of the Uniform Rules of Conduct" [UN90]. According to the ontology six patterns are defined for the business transactions: commercial transaction, query/response, request/response, request/confirm, information distribution and notification. These business transaction activity patterns comprehensively cover all known legally binding collaborations at the level of request/response and one-way interaction between two business applications. These patterns are successfully implemented in the RosettaNet Implementation Framework.

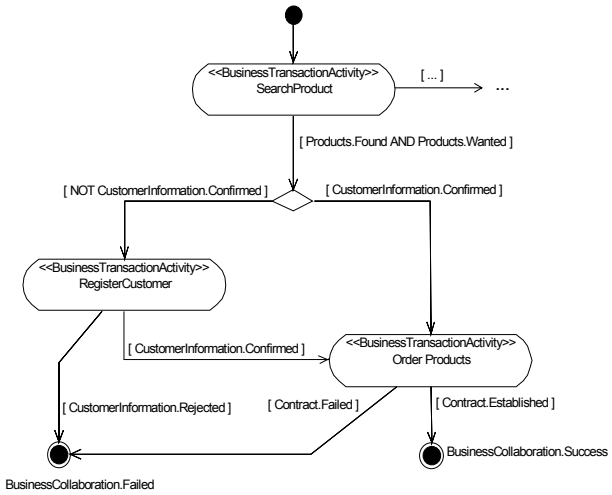

Fig. 8. Business Collaboration Protocol

Fig. 9 details the register customer transaction which follows the pattern of a Request/ Response transaction. The transaction is built by the initiating activity request registration performed by the buyer and the responding one register customer performed by the seller. The business information exchanged is a registration request and a registration response. Furthermore, all the messaging and security parameters for the activities and the business information are defined. The information exchange is commonly called a business document. Since the BCF does not represent a document-centric approach and only exchanges information necessary for a business state change, we stick to the term business information. The information structure for the registration request is depicted in the class diagram on the right-hand side of Fig. 9. The information structure usually covers general information of the request/response and is composed of the business entities that are effected by the transaction. Each business entity is described by the information needed to change its business state. This information is built by re-using business objects in a B2B library. Thus, one must select suitable business objects from this library and customize them to the needs of the business transaction. Customizing means setting the business objects into the context of the business transaction. Note, that we use the terms business entity and business objects as defined in UMM - others would prefer to call them the other way round.

5.4 Business Service View

The fundamental principle of the business service view (BSV) is to describe the business collaborations between network components. The reference ontology for the BSV comes from the "UN/ECE Recommendation No. 26, The Commercial Use of Interchange Agreements for Electronic Data Interchange". A total of 24 service interaction patterns have been identified. They specify interaction sequences between two application systems, i.e., protocols of message exchanges, according to the type of business transaction, type of role, security and timing parameters.

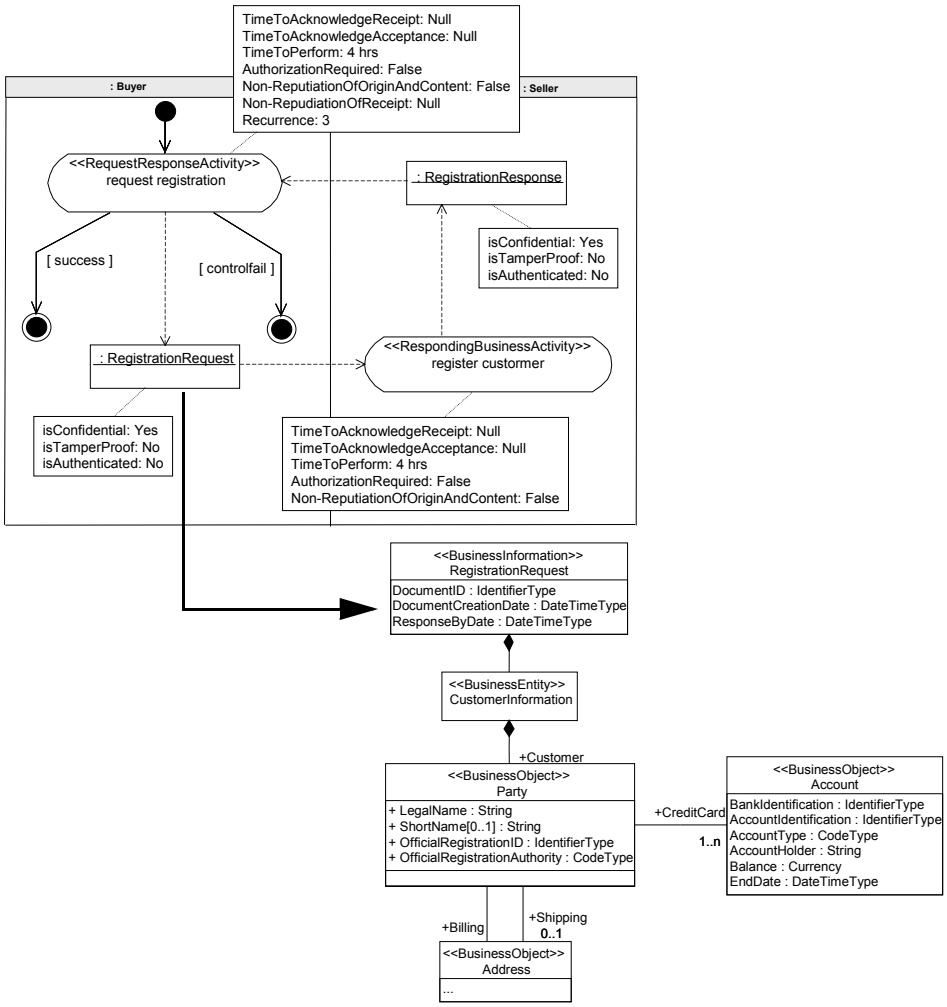

Fig. 9. Business Transaction Register Customer and Information Structure for the Registration Request

The business service workflow does not add any new information. Accordingly, the business service view artifacts are automatically created from the information gained in the previous workflows. Each business transaction of the BTV maps to a corresponding service transaction in the BSV. Fig. 10 depicts the register customer service transaction that follows a service-service pattern. The corresponding business transaction does not require any acknowledgments. Therefore, the resulting sequence diagram is quite simple and only includes document exchanges. The buyer service calls the register customer operation of the seller service by sending the search request document. The seller service returns the search result document to the request a search operation that initiated the service transaction. By mapping all five business transactions of our example to service transactions, we exactly define the services to be supported by each network component.

Fig. 10 denotes the resulting network components as objects and their services as operations. Software vendors supporting a role in the simple order management must implement the corresponding interface class in their applications.

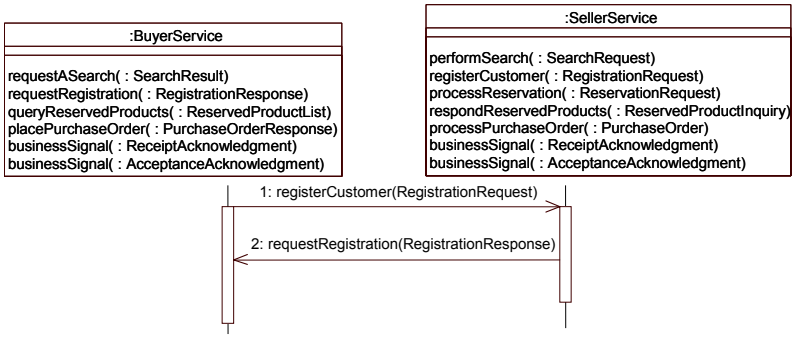

Fig. 10. Business Service Transaction

6 Summary

The Web Services standards and ebXML ensure interoperability at the technology layer of e-business applications. They do not guarantee interoperability at the business layer. This paper presents UN/CEFACT's Business Collaboration Framework (BCF) which provides a solution to define business collaborations independent of the underlying implementation and infrastructure technology. The BCF delivers unambiguous specifications for the collaboration's choreography as well as for the information exchanged based on business state changes. It sets all business semantics that must be covered by Web Services standards and ebXML to become machine-processable. The combination of BCF and Web Services standards and/or ebXML will lead to the full potential of today's e-business. To reach the envisioned goal two steps are necessary: Firstly, a methodology for transforming BCF artifacts into Web Services standards and ebXML must be developed. Secondly, one must convince the COTS software providers to develop B2B software based on the most popular BCF scenarios.

7 References

[BJR98] Booch, G., Jacobson, I., Rumbaugh J.: The Unified Modeling Language User Guide. Addison Wesley Object Technology Series, Reading, (1998)

[Eu02] Commission of the European Communities: Draft amending Commission Recommendation 96/280/EC concerning the definition of small and medium-sized enterprises. Brussels (2002), http://europa.eu.int/comm/enterprise/consultations/sme_definition/consultation2/index_en.htm

[EN01] Eisenberg, B., Nickull, D. (Eds.): ebXML Technical Architecture Specification v1.0.4. (2001), http://www.ebxml.org/specs/ebTA.pdf

[Fe00] Feldman, S.: The Changing Face of E-Commerce: Extending the Boundaries of the Possible. IEEE Internet Computing, Vol. 4, No. 3; 2000

[FF03] Ferris C., Farrell J.: What are Web Services. Communications of the ACM, Vol. 46, No. 6; 2003

[HF89] Hill, N.C., Ferguson, D.M.: Electronic Data Interchange: A Definition and Perspective. EDI Forum: The Journal of Electronic Data Interchange, Vol. 1, Issue 1, pp. 5-12, (1989)

[HHK02] Hofreiter, B., Huemer, C., Klas, W.: ebXML: Status, Research Issues and Obstacles. Proc. of 12th Int. Workshop on Research Issues on Data Engineering (RIDE02), San Jose (2002)

[HH03] Hofreiter, B., Huemer, C.: Modeling Business Collaborations in Context. Proc. of On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, Springer LNCS, Catania (2003)

[Hu01] Huemer, C.: <<DIR>>-XML² - Unambiguous Access to XML-based Business Documents in B2B E-Commerce. Proc. of 3rd ACM Conference on Electronic Commerce, Tampa (2001)

[ISO95] ISO: Open-edi Reference Model. ISO/IEC JTC 1/SC30 ISO Standard 14662 (1995)

[Li00] Li, H.: XML and Industrial Standards for Electronic Commerce. Knowledge and Information Systems, Vol. 2, No. 4 (2000)

[Mc03] McCarthy W.E.; REA Enterprise Ontology Source Page; http://www.msu.edu/user/mccarth4/rea-ontology/

[PN03] Patil S., Newcomer E.; ebXML and Web Services. IEEE Internet Computing, Vol. 7, No. 3, 2003

[Sc88] Schatz W.; EDI: Putting The Muscle In Commerce And Industry. Datamation, Vol. 34, No. 6; March 1988

[UCC92] Uniform Commercial Code Committee: The Commercial Use of Electronic Data Interchange - A Report and Model Trading Partner Agreement, 1992

[UN90] UN/ECE: Uniform Rules of Conduct for Interchange of Trade Data by Teletransmission (UNCID); http://www.unece.org/trade/untdid/texts/unrci.htm

[UN03] UN/CEFACT TMG: UN/CEFACT Modelling Methodology, Revision 12. http://www.untmg.org

[WW02] W3C Web Services Architecture Working Group; Web Services Architecture Requirements; W3C Working Draft; Nov. 2002; http://www.w3.org/TR/2002/WD-wsa-reqs-20021114