The Development Process of the UN/CEFACT Modeling Methodology

Christian Huemer
Business Informatics Group
TU Vienna
huemer@big.tuwien.ac.at

Philipp Liegl
Business Informatics Group
TU Vienna
liegl@big.tuwien.ac.at

Thomas Motal
Electronic Commerce Group
TU Vienna
motal@ec.tuwien.ac.at

Rainer Schuster
Electronic Commerce Group
TU Vienna
schuster@ec.tuwien.ac.at

Marco Zapletal
Electronic Commerce Group
TU Vienna
marco@ec.tuwien.ac.at

ABSTRACT

The development of inter-organizational systems requires a well defined development process. UN/CEFACT's Modeling Methodology (UMM) provides such a development process. We served as the editing team of the UMM 1.0 foundation module, which is defined as a UML profile. First experiences of applying UMM in real world projects have disclosed some limitations. Accordingly, we propose integrating new concepts into a new version 2.0 of UMM. In this paper, we show the adapted UMM development process, which is demonstrated by means of a waste management example.

Categories and Subject Descriptors

J.1 [**Computer Applications**]: Administrative Data Processing—*Business*; D.2.11 [**Software Engineering**]: Interoperability

General Terms

Inter-organizational Business Process Modeling

Keywords

Business Process Modeling, Inter-organizational Systems, UN/CEFACT's Modeling Methodology, B2B Standardization

1. INTRODUCTION

The United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT) is an international e-business standardization body known for its work in the area of electronic data interchange (EDI). UN/CEFACT maintains the UN/EDIFACT standards and participated in the ebXML initiative. However, in the recent years the world of EDI has changed significantly. The pure document-centric approach is going to be replaced by a service-oriented one. In this context, services are economic activities offered to other business partners in order to achieve a certain benefit [50]. Realizing the services portfolio in a technical sense results in B2B information systems according to the concept of a service-oriented architecture (SOA).

Web Services are the state-of-the-art technology for implementing a SOA. Evidently, a successful B2B integration does not start with manually creating Web Services artifacts, such as WSDL or BPEL code. Successfully implementing a SOA requires an evolutionary development process which is similar to object-oriented and component-based software development processes and which considers business process modeling as a driving factor (cf. [31]).

For analyzing and designing inter-organizational systems, UN/CEFACT has started to work on a development process called UN/CEFACT's modeling methodology (UMM) in 1998. During the course of time this development process has changed considerably. UN/CEFACT's Modeling Methodology (UMM) is an integrated approach for capturing the collaborative space between organizations. The UMM has always been developed according to the business operational view of the Open-edi reference model [12], which covers the business aspects such as business information, business conventions, agreements, and rules among organizations.

When UN/CEFACT and OASIS started the ebXML initiative in 1999, UMM concepts have significantly influenced the ebXML business process specification [40]. Also UMM changed during this time by adopting concepts from ebXML members, such as SWIFT, TM Forum, GS1 (EAN*UCC), and RosettaNet. In 2000 the copyrights of the company EDIFECS on their Business Collaboration Framework (BCF) used by RosettaNet were transferred to UN/CEFACT and the BCF was merged into UMM.

However, at this time UMM just provided guidelines for using the general purpose modeling language UML [29]. It missed a formal customization of the UML meta model. Furthermore, we recognized a step toward service orientation. Being part of the UMM project team, we addressed these challenges by developing a UML profile that integrates service-oriented concepts. A UML profile specifies a set of stereotypes, tagged values and constraints for customizing UML. This means the general-purpose language UML is customized for the specific purpose of inter-organizational systems. Thereby, UMM puts UML in a very strict corset. The resulting artifacts are well defined. Each artifact is restricted to a number of precisely defined modeling elements (stereotypes) and the relationships among them is also fixed. As a consequence, it is easier for software engineers to act upon the resulting artifacts in order to bind their local systems to the public process defined by UMM. We have been the editing team of the resulting specification 'UMM foundation module 1.0' [41] [8] that was finalized in 2006. Our book [20] provides an extensive user guide on UMM 1.0. A sur-

10th Int. Conf. on Electronic Commerce (ICEC) '08 Innsbruck, Austria

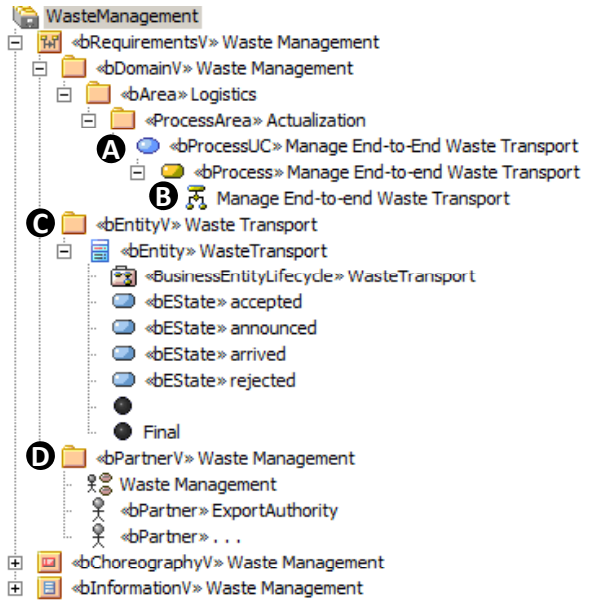

Figure 1: Overview of the Business Requirements View

vey of different B2B business process modeling languages [2] has shown, that UMM 1.0 is the most complete approach.

However, first experiences in applying the UMM in real world projects have shown some shortcomings: First, the current UMM provides rather vague means for modeling business documents. Second, there is a lack of alternative responses in a business transaction. Third, results of a business transaction currently do not propagate changes of business entity states. Fourth, current UMM business choreographies used guards in natural language and, thus, lack information to be machine-processable. Fifth, UMM does not allow to interlink activities of two different business collaborations. Finally, stakeholders have argued against the complex package structure of a UMM 1.0 model.

Consequently, we propose new concepts to be adapted by UMM in order to overcome the limitations mentioned above. We submitted these concepts to UN/CEFACT in order to move the UMM foundation module toward version 2. The goal of this paper is to demonstrate the adapted UMM 2.0 development process which overcomes the limitations of UMM 1.0. We go step by step through the development process of a rather simple, but still realistic example. This example is taken from a project in the European `waste management` domain. Crossborder transports of waste - even within the EU - are subject to regulations. A transport must be announced and the receipt of the waste as well as its disposal must be signaled. Exporter, importer, and the relevant authorities in their countries interchange this information.

In the following sections we go through the three main sections of the UMM 2.0: *business requirements view (bRequirementsV)*, *business choreography view (bChoreographyV)* and *business information view (bInformationV)*. Note, throughout the paper the stereotype names are shown in parentheses, which are abbreviated forms of the views' full names. However, in the text we use the full name.

2. BUSINESS REQUIREMENTS VIEW

The business requirements view is the first view to be constructed during the elaboration of a UMM model. Figure 1 shows the package structure of the *business requirements view* and its three subviews *business domain view (bDomainV)*, *business entity view (bEntityV)*, and *business partner view (bPartnerV)*. The alphabetically numbered dots associate the example diagrams with the respective packages they belong to, e.g. figure 3 shows the detailed view of A in figure 1.

Form: *BusinessProcess*	
General	
Business Process Name	Manage End-to-End Waste Transport
Definition	A waste transport taking place between an export authority and an import authority.
Description	Subject of the business process is the waste transport between different countries. The export authority of the export country pre-informs the import authority of the import country about a waste transport. Upon successful receipt of the waste transport the import authority informs the export authority.
Participants	ImportAuthority, ExportAuthority
Stakeholder	Tax Agency
Reference	Waste Management
Start/End Characteristics	
Pre-condition	The waste is ready for transport.
Post-condition	- The waste has been moved from the export country to the import country. - No waste transport took place.
Begins When	Export authority receives the order to initiate the waste transport.
Ends When	The export authority receives the transport arrival receipt from the import authority.
Actions	- Pre-inform on waste transport - Inform on waste receipt
Exceptions	-
Relationships	
Included Business Processes	none
Affected Business Entities	WasteTransport

Figure 2: Business Process Worksheet

2.1 Business Domain View

At the beginning of the UMM development process, the business analyst gathers domain knowledge and existing process knowledge of the business domain under consideration. The analyst has to capture the justification of the project and has to determine its scope. He interviews business experts and other stakeholders to get an understanding of the existing business processes in the domain. Thereby, worksheets are a popular mechanism to guide the interview and to capture business know-how. Worksheets are structured forms for the elicitation of specific requirements. It is important that the analyst does not influence the business expert. The interview has to take place in the language of the business domain expert; technical and modeling terms should be avoided. The interviews ensure that all involved parties share a common understanding of the business domain. In this step, the analyst discovers intra- and inter-organizational business processes as existing or desired by individual parties. A simplified example for the output of an interview kept in a worksheet is depicted in figure 2.

The results of the interviews are transformed into a UML notation. Each worksheet describing a business process results in a *business process uses cases (bProcessUC)*. Business processes are classified according to UN/CEFACT's Catalog of Common Business Processes (CBPC) [42], the Supply Chain Reference Model (SCOR) [39] or Porter's Value Chain (PVC) [33]. Classifying business processes facilitates the understanding of the business domain as well as its scope. A hierarchical composition of business areas and process areas is used to represent the classification as shown in figure 1. In this example we only show one *business area* `logistics` which includes the *process area* `actualization`. In reality, a *business domain view* comprises additional *business* and *process areas*.

The business process use case manage end-to-end waste transport is assigned to the process area actualization within the business area logistics (A in figure 1). The corresponding use case diagram is shown in figure 3. In general, business partners participating in the business processes and stakeholders who have an interest

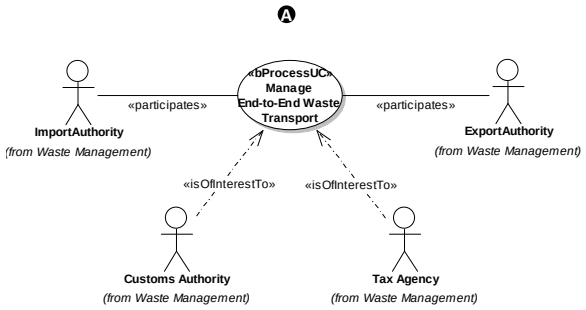

Figure 3: Business Process Use Case with Business Partners

in them are associated to the business process use cases. In our example, the *business partners* `exporter`, `export authority`, `import authority`, and `importer` participate in `manage end-to-end waste transport`, whereas the `stakeholders` `customs authority` and `tax agency` have an interest.

Once all business processes are discovered, a review cycle is initiated in order to identify those who in fact have a relevance for the business collaboration to be developed. These business process are further detailed by an activity diagram according to the requirements specified in the respective worksheet. The activity diagram becomes a child of the *business process use case*. In our example, we show the activity diagram for manage end-to-end waste transport in figure 4. According to figure 1, this activity diagram (B) is a child of the corresponding business process use case (A).

The following business semantics are kept in the activity diagram: An `exporter` informs the `export authority` about a waste transport. The `export authority` in turn informs the `import authority` about the incoming waste transport. The `import authority` then informs the `importer`. The flow of accepting or rejecting the waste transport is going into the reverse direction. In case the waste transport announcement has been accepted the waste transport starts. Upon arrival of the waste in the import country, the flow of informing partners on its receipt is also going the reverse direction. Due to space limitations, we only show the activities between the export authority and the import authority in detail, whereas the other activities are only rudimentarily highlighted.

The exchange of information must always lead to a synchronization of changed *business entity states* at each partner's side. Thus, the object flow between activities is denoted by a *shared business entity state*, which is further discusses below in the subsection on the business entity view. The concept of *shared business entity states* denotes the need for communication between business partners. Thus, *shared business entity states* are a strong indicator for requiring information exchange in later designed business collaborations.

2.2 Business Entity View

A *business entity* is a real-world thing having business significance that is shared between two or more business partners in a collaborative business process (e.g. "order", "account", etc.). In our example, the information exchanged is about the *business entity* `waste transport`.

A *business entity lifecycle* is described by a UML state diagram as part of the business entity view (cf. C in figure 1). It delineates the states a *business entity* may obtain as well as the flow between them. The lifecycle is designed in accordance with the activity diagrams in the *business domain view*. The object flow in the activity

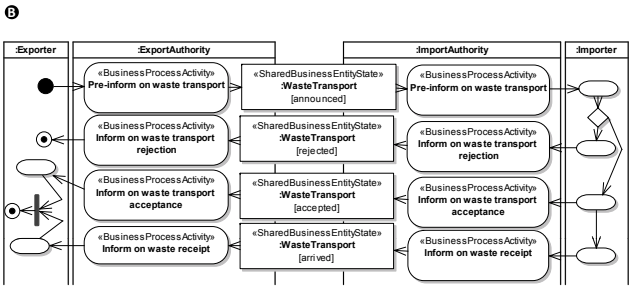

Figure 4: Business Process Activity Model

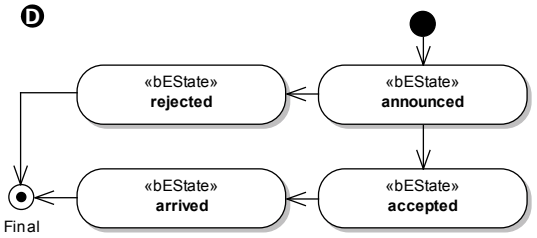

Figure 5: Business Entity Life Cycle: Waste Transport

diagrams is based on *shared business entity states* (cf. figure 4). Each *shared business entity state* reflects a *business entity state* in the *business entity lifecycle* (cf. figure 5). Thus, the order of changing *business entity states* in the activity diagrams must be kept in the *business entity lifecycle*.

The *business entity lifecycle* depicted in figure 5 represents the states of the *business entity* `waste transport`. It is created with state `announced`. The pending state `announced` is either set to `approved` or `rejected`. After the approved transport happened, the *business entity* is set to `arrived`. These four *business entity states* are referenced by the four *shared business entity states* of the activity diagram in figure 4.

Business partners identified in the business requirements view are modeled in diagrams that belong to the business domain view. However, for the sake of an easier re-use, business partners and stakeholders are kept in a dedicated container - the *business partner view* ((D) in figure 1). The business partner view may also be used to analyze relationships between the business partners and/or stakeholders in optional role models, which are not further elaborated here.

3. BUSINESS CHOREOGRAPHY VIEW

In the *business choreography view* the analyst builds upon the previously created artifacts in order to develop models describing a global choreography. According to figure 6, it consists of three subviews: *business transaction view (bTransactionV)*, *business collaboration view (bCollaborationV)* and *business realization view (bRealizationV)*. The *business transaction view* models the basic-building blocks of a choreography which correspond to a single business document exchange and returning an optional business document as a response. The *business collaboration view* models a global choreography built by these basic building blocks. A *business realization view* is used if the same choreography is realized between different set of business partners.

3.1 Business Transaction View

The basic building blocks of a UMM choreography are *business*

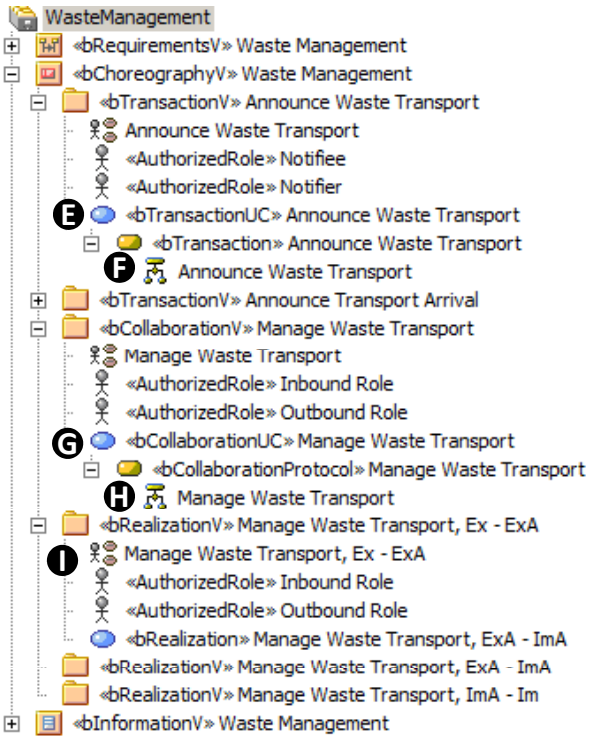

Figure 6: Overview of the Business Choreography View

transactions. The goal of a business transaction is synchronizing the business entity states between two parties. Synchronization of states is either required in an uni-directional or in a bi-directional way. In the former case, the initiator of the *business transaction* informs the other party about an already irreversible state change the other party has to accept - e.g., the notification that the waste has arrived. It follows, that responding in such a scenario is neither required nor reasonable. In the latter case, the initiating party sets a *business entity* to an interim state and the responding party decides about its final state - consider a request for a waste transport that the responder might either accept or refuse.

The synchronization takes place by exchanging business information. According to the definitions above, an exchange takes always place between exactly two parties. It is either a uni-directional exchange or a bi-directional exchange including a response. The activity diagrams created in the business domain view (cf. 4) already indicate the need for exchanging business information to synchronize business entities by the concept of *shared business entity states*. However, these activity diagrams are not necessarily consolidated between the various parties and are just used for requirements elicitation. The business transaction has to present a consolidated view on the basic building blocks. Thus, it has to identify the commonly agreed *shared business entity states* and, possibly, aggregate two of them in a bi-directional business information exchange.

This identification and consolidation process leads to a number of *business transaction use cases* and the two *authorized roles* participating in the use case. According to figure 6, each *business transaction use case* (E) and the two participating *authorized roles* are placed in their own *business transaction view*. Figure 7 depicts the *business transaction use case* `announce waste transport` which involves the participating *authorized roles* `notifier` and

Figure 7: Business Transaction Use Case

`notifiee`. Note, we use the abstract concepts of *authorized roles* instead of *business partners*, because *business transactions* and their use cases may be realized between different sets of *business partners*.

The requirements of a *business transaction* are further elaborated using the concept of an activity diagram. For each *business transaction use case* an activity diagram is created and placed as a child underneath the respective use case, e.g., in figure 6 the *business transaction use case* `announce waste transport` (E) is refined using the activity diagram (F).

The main purpose of a business transaction activity diagram is to formally describe a UMM business transaction. It is important to notice, that a *business transaction* always follows the same pattern. The business transaction pattern thereby defines the type of a legally binding interaction between two decision making applications as defined in Open-edi [12]. We distinguish between two one-way (information distribution, notification) and four two-way (query/response, request/response, request/confirm, commercial transaction) types of business transactions.

The basic building blocks of a business transaction are activity partitions, which are used to denote the authorized roles, participating in the transaction. Furthermore, a business transaction contains exactly two actions - a requesting action and a responding action - one on each business partner's side. Between the different actions the business information exchange is denoted using the concepts of object flows and action pins. There is always exactly one object flow from the requesting action to the responding action. In a one-way business transaction there is no flow in the reverse direction. In case of a two-way business transaction there are one or more object flows in the reverse direction. In case of two or more object flows they are considered as alternatives. The type of the action pins in the business transaction is set using business documents from the business information view (see section 4).

Figure 8 shows the *business transaction* `announce waste transport`. On the left hand side the *business transaction partition (bTPartition)* of the requesting role is shown and on the right hand side the one of the responding role. The type of a *business transaction partition* is determined by the *authorized roles* participating in the *business transaction use case*, which the *business transaction* refines. In figure 8 the type of the requesting partition is set by the *authorized role* `notifier` and the type of the *responding partition* is set by the *authorized role* `notifiee`.

The *requesting partition* contains a so called *requesting action (ReqAction)* and the *responding partition* a *responding action (ResAction)*. In the example shown in figure 8 the `notifier` starts the *business transaction* by sending a `waste movement form` to the `notifiee`. Since the transaction is bi-directional the *business entity* `waste transport` is set to an interim state. Depending on the response of the `notifiee`, the *business entity* is set to its final state.

After the `notifiee` has processed the request from the `notifier` he either replies with a `waste movement accepted form` or with a `waste movement rejected form`. In the notifier's partition two *shared business entity states* `waste transport` are shown to-

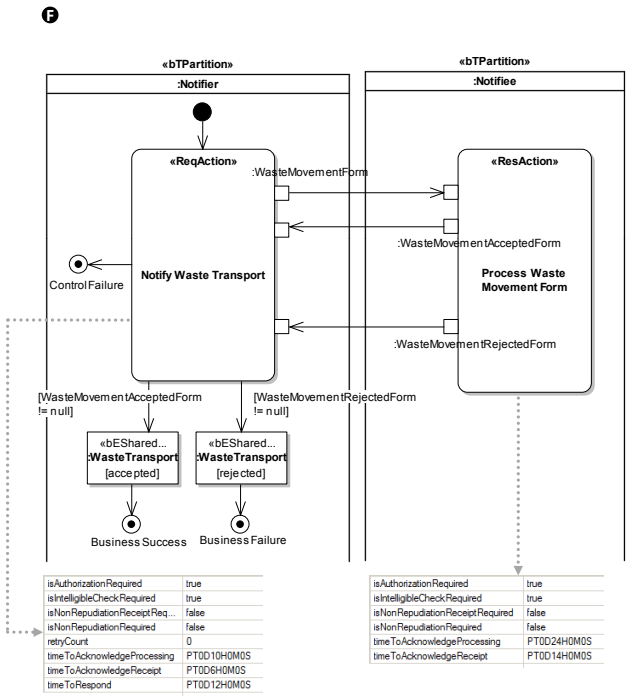

Figure 8: Business Transaction *Announce Waste Transport*

gether with guard conditions leading to the *shared business entity states*. Depending on the reply of the `notifiee` the *shared business entity state* `waste transport` is either set to the final state `accepted` or to `rejected`. In case a control failure occurs during the transaction the *business transaction* results in a control failure as shown on the left hand side of figure 8.

At the lower side of figure 8 the tagged values containing the different business signal information of the requesting and the responding action are shown e.g. time to acknowledge receipt indicates the maximum time within the responding party has to confirm a successful/unsuccessful syntax, grammar, and sequence validation. Further tagged values are: is authorization required, is non-repudiation required, time to perform, time to acknowledge receipt, time to acknowledge acceptance, is non-repudiation of receipt required and retry count. These tagged values are shown at are considered as self-explanatory and are explained in detail in the UMM 1.0 specification [41].

As shown in figure 6 the waste management example consist of exactly two business transactions: *announce waste transport* (figure 8) and *announce transport arrival*. The latter one is a one-way transaction and is not explained in detail here.

3.2 Business Collaboration View

After the identification of the different *business transactions* the modeler continues with creating *business collaborations*. A *business collaboration* choreographs the execution order of different *business transactions* and *business collaborations* (since *business collaborations* can be recursively nested).

Each *business collaboration view* contains exactly one *business collaboration use case* and two *authorized roles* participating in the use case (E in figure 6). By definition a *business collaboration* consists of different *business transactions* and/or *business collaborations*. Included *business transactions/collaborations* are denoted using the concept of *include* dependencies. Each included *business transaction* is defined in its own *business transaction view* and each included *business collaboration* is defined in its own *business collaboration view*.

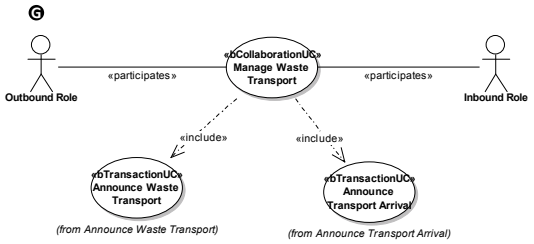

Figure 9: Business Collaboration Use Case

As shown in figure 9 the *business collaboration use case* `manage waste transport` includes two *business transactions*, namely `announce waste transport` and `announce transport arrival`. Again the abstract concept of *authorized roles* is used instead of *business partners* because *business collaborations* may be realized between different sets of *business partners*.

Similar to the concept of a *business transaction use case* a *business collaboration use case* is further elaborated using the concept of a so called *business collaboration protocol*. For each *business collaboration use case* a *business collaboration protocol* is created and placed as a child under the respective use case e.g. in figure 6 the *business collaboration use case* `manage waste transport` (G) is refined using the *business collaboration protocol* (H). Consequently a *business collaboration use case* is always the parent of exactly one *business collaboration protocol*.

The main aim of a *business collaboration protocol* is to describe a *business collaboration* on a formal basis. Thereby, a *business collaboration protocol* is built using *business transaction actions* and *business collaboration actions*. A *business transaction action* calls a *business transaction* and a *business collaboration action* calls a *business collaboration*. In order to depict the *authorized roles* participating in a *business collaboration*, a *business collaboration protocol* uses the concept of partitions. For each *authorized role* exactly one partition is created. In some cases an *authorized role*, during the course of a *business collaboration*, might internally execute another *business collaboration*. In this case the concept of so called *nested business collaboration* is used. *Nested business collaborations* are defined in another *business collaboration view*. In order to denote the execution order of different *business transaction actions* and *business collaboration actions* the concept of *initFlows* and *reFlows* is used. Thereby an *initFlow* can either lead to a partition or - in case a *nested collaboration* is used - to a *nested business collaboration*. The same applies to *reFlows*. Guard conditions attached to the different object flows within the *business collaboration protocol* determine the exact execution sequence.

The *business collaboration protocol* in figure 10 defines the exact choreography of the `manage waste transport` collaboration. Using the concept of two *business collaboration partitions (bCPartition)* the two *authorized roles* `outbound role` and `inbound role` participating in the `business collaboration` are shown. The *business collaboration* `management waste transport` starts with the *business transaction* `announce waste transport`. The *initFlow* dependency between the `outbound role` and the *business transaction action* `announce waste transport` in figure 10 indicates, that the `outbound role` initiates the *business transaction*. Since there is a *reFlow* dependency from the *nested business collaboration* within the partition of the `inbound role` to

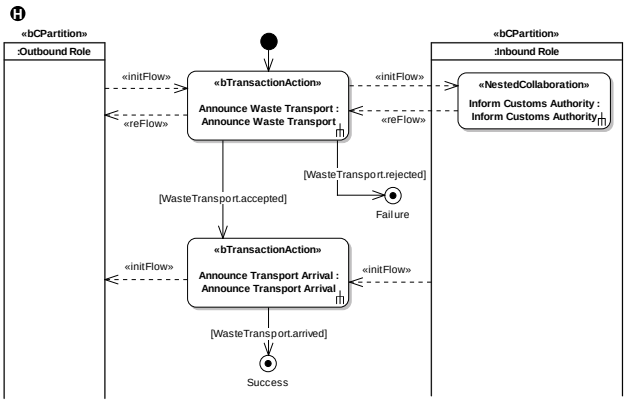

Figure 10: Business Collaboration Protocol *Manage Waste Transport*

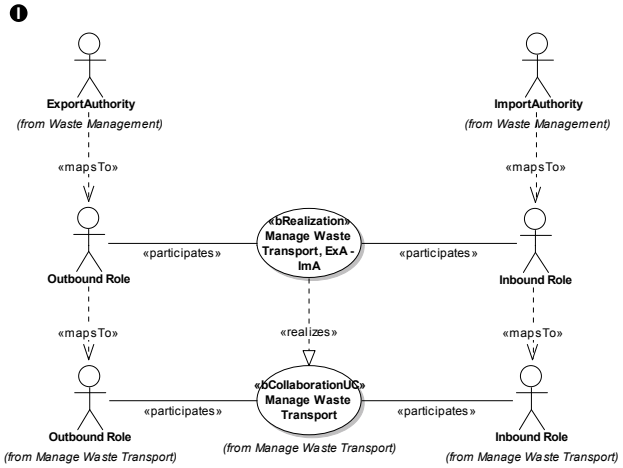

Figure 11: Business Realization View

the *business transaction action* and the `outbound role`, the *business transaction* is a two-way transaction. The `inbound role` informs the `customs authority` about the waste transport announcement of the `outbound role`. If the `customs authority` rejects the waste transport, the `inbound role` rejects the waste transport as well and sends a `waste movement rejected form` to the `outbound role`.

If the *business transaction* `announce waste transport` fails, because the `inbound role` or the `customs authority` has rejected the transport, the *business collaboration* `manage waste transport` also fails. In figure 10 this is indicated by the control flow with the guard condition `WasteTransport.rejected` leading from the *business transaction action* to the final state `Failure`. Please note, that the guard conditions of the control flows directly match to the *shared business entity states* of the underlying *business transaction* (see figure 8).

In case the *business transaction* `announce waste transport` was successful, the guard condition `WasteTransport.accepted` evaluates true and the *business transaction* `announce transport arrival` starts. Please note that now the `inbound role` is the initiator of the *business transaction*. The `inbound role` has received the waste from the `outbound role` and now informs the *business partner* about this irreversible state. As shown in figure 10 this is indicated by the *initFlow* dependency between the `inbound role` and the *business transaction action* `announce transport arrival`. The *business collaboration* finally ends with the *business entity* `waste transport` being in state `arrived`.

3.3 Business Realization View

We have seen so far, that *business transactions* and *business collaborations* are executed between *authorized roles* instead of specific *business partners*. By using the concept of authorized roles, the same business collaboration/transaction may be re-used between different sets of specific business partners. This enables the standardization of business collaboration models and, in turn, fosters re-use, which is one of the key goals of UN/CEFACT.

In order to bind a *business collaboration* (and implicitly the *business transactions* it consists of) to a set of *business partners*, UMM provides the concept of so called *business realizations*. Figure 11 shows a possible *business realization* for the *business collaboration* `manage waste transport`.

On the lower left hand side of figure 11 the *business collaboration* `manage waste transport` is shown between the two *authorized roles* `outbound role` and `inbound role`. A *business realization* is connected to a specific *business collaboration use case* using a *realize* connection. In figure 11 the *business realization* `manage waste transport ExA-ImA` *realizes* the *business collaboration use case* `manage waste transport`. The *business realization* again has two *authorized roles* `outbound role` and `inbound role`. Finally, *business partners* identified in the *business partner view* are bound to *authorized roles* by connecting them via *mapsTo* dependencies.

The benefit of this concept is easily demonstrated by our example. As we learned in section 2.1, our example business collaboration between `export authority` and `import authority` is identical to the one performed between `exporter` and `export authority` as well as to the one between `import authority` and `importer`. This issue is modeled by introducing two additional business realizations, which both realize the *business collaboration use case* `manage waste transport`. One of them is performed between the `exporter` and the `export authority` and the other one between the `import authority` and the `importer`. Thus, the concept of *business realizations* evidently contributes to the reuse of modeling artifacts.

With the completion of the *business realization view* the business modeler has finished the business process perspective of the UMM.

4. BUSINESS INFORMATION VIEW

The final view of UMM is the so called *business information view*. Within the *business information view* the business documents, which are exchanged in the different *business transactions* of UMM are defined. UMM does not mandate to use a specific business document modeling technique in this view, but leaves it up to the modeler which technology to use. It is, however, strongly suggested to use UN/CEFACT's Core Components [43] for the modeling of the exchanged business documents. Core components are syntax independent, reusable building blocks, standardized by UN/CEFACT for the modeling of business documents. In order to allow for an integration of core components into a UML modeling tool, UN/CEFACT has developed the UML Profile for Core Components (UPCC) [44].

With the help of UPCC *core components* and so called *business information entities* are modeled. Thereby *core components* are context independent and generic building blocks for business documents. If a *core component* is used in a certain business scenario it becomes a so called *business information entity*. A *business in-*

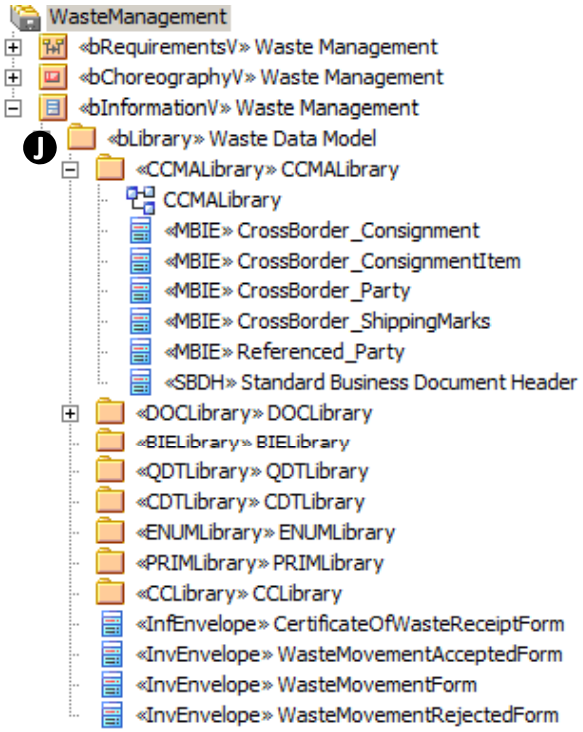

Figure 12: Overview of the Business Information View

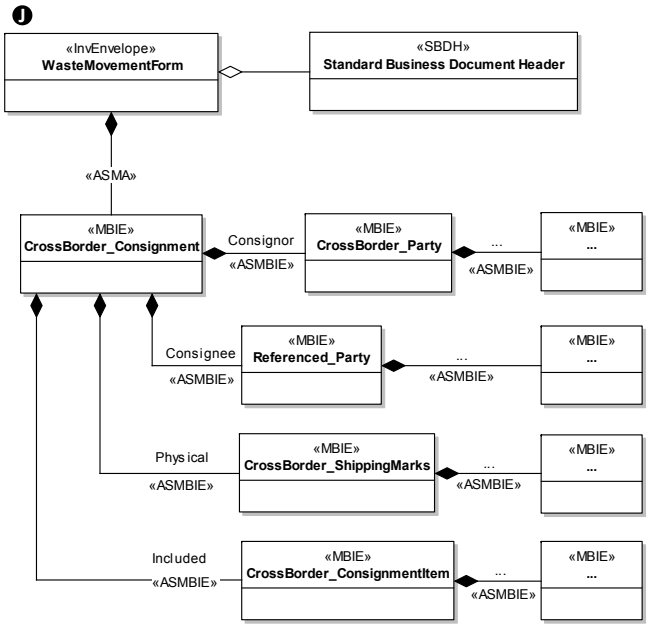

Figure 13: Business Information View Example

formation entity has a certain context and is always derived from the underlying *core component* by restriction. Hence, the generic *core component's* building blocks are tailored to the specific needs of a certain business scenario.

The second standard to be used within the *business information view* is the so called Core Component Message Assembly (CCMA) [45]. Within the core component message assembly so called *business messages* are defined. *Business messages* are used to update business documents. The reason therefore is, that not all of the business document information has to be exchanged whenever a business transaction is initiated. For example, not every time the arrival of a waste transport is communicated to the `export authority`, the `import authority` has to send all the waste transport details. Instead the exchange of an identification, referencing the waste is sufficient.

Figure 12 shows the structure of the *business information view* which consists of a *business library (bLibrary)* called `waste data model`. Within a *business library* the modeler aggregates the different elements of the UML Profile for Core Components (UPCC) and core component message assembly (CCMA) standard.

At the bottom of figure 12 four different *information envelopes* used in the `manage waste transport` collaboration are shown. Above the *information envelopes* several packages are shown, which belong to the UPCC and CCMA standards. Exemplarily the `waste movement form` is examined in detail.

As shown in figure 13 the *information envelope* `waste movement form` consists of a *standard business document header* and a *messaging business information entity (MBIE)* `cross border consignment`. The `cross border consignment` is connected to the *information envelope* using the concept of an *association message assembly (ASMA)*. A `cross border consignment` consists of four additional *messaging business information entities*. *Messaging business information entities* are connected to each other using the concept of *association messaging business information entities (ASMBIE)*. In the example shown in figure 13 the four ASMBIEs are `consignor`, `consignee`, `physical`, and `included`. Due to space limitations the referenced *messaging business information entities* are not shown in detail.

Finally, the different *information envelopes* defined in the *business information view* are used to set the type of the outgoing action pins of a requesting and responding action in a *business transaction*. (e.g. compare J in figure 12 with figure 8).

5. RELATED WORK

Over the last couple of years, a lot of methodologies for modeling business processes have been developed. Surveys comparing different types of business process modeling languages are provided in [21] [15]. Some of these approaches are based on special notations [37] [24] [28] [48]. Others customize the UML for business process modeling needs [23]. Most of these UML approaches are based on activity diagrams [36] and they either provide just guidelines on using activity diagrams for this special purpose or they specify a UML profile. Traditionally, business process modeling focuses on modeling business processes internal to an organization fulfilling customer needs [32]. More recent approaches also take inter-organizational business processes into account [14] [16]. Due to the growing importance of XML and Web Services, a lot of XML-based notations describing the orchestration and choreography of executable business processes have been developed. The most popular languages in this area are the Business Process Execution Language for Web Services (BPEL4WS) [27] [19] and the Business Process Specification Schema (BPSS) [40]. Solutions for a straight-forward transformation of UMM business transaction models already exist [9] [6] [7] [11].

A business process modeling approach famous for requirements modeling are so called Event-driven Process Chains (EPCs). EPCs specifically focus on the control flow dependencies between the different functions within a business process. They are therefore well suited to depict the dynamic behavior of a specific process. If

the user however wants to extend the information stored in an EPC with static information little or no support is provided. Mostly such documents, similar to use case descriptions, are stored separately to the model - hence lowering maintainability and traceability of requirements information. EPCs are utilized in the ARchitecture of Integrated Information Systems (ARIS), a methodology developed by Scheer [37]. Within ARIS Event-driven Process Chains are the central method for the conceptual integration of the functional, organizational, data and output perspective in the information system's design.

Another non-UML based approach is the so called Integrated DEFinition Method 3 (IDEF3) [24] which is used to model the sequential behavior of processes and systems. IDEF3 supports two different types of models: a process flow description and an object state transaction description. Both EPCs and IDEF3 do not have a meta model such as UML based modeling languages.

A more formal and mathematical approach to business process modeling is provided by Petri-Nets [26]. Petri-Nets are used to model both, business processes [48] as well as workflow systems [47]. Several approaches are using Petri-Nets for the modeling of inter-organizational processes. Lee [17] has contributed an implementation of the choreography aspects of ISO's Open-edi reference model [12]. Several other approaches have been taken up by different authors in order to use Petri-Nets for inter-organizational modeling [18] [22] [46]. One major problem a modeler using Petri Nets is facing, is the increasing complexity of a Net if complicated business processes are modeled. Hence its applicability in the field of requirements elicitation is rather limited since the resulting net becomes illegible for non-technicans.

A more recent approach is the so called Business Process Modeling Notation (BPMN) [28] being developed by the Object Management Group. One major goal of the BPMN initiative is to create one single graphical modeling notation understandable by as many stakeholders as possible - from the business analyst to the application developer. Direct mapping between graphical-based notations and block-formatted languages - such as BPEL - is a challenging task [49]. A first attempt toward an automated BPMN2BPEL mapping has been outlined in [34]. The generation of BPEL code will be maintained by a pattern-algorithm discovering BPMN patterns within the graph that can be mapped to BPEL structured fragments.

The advantage of a business based requirements approach has also been outlined in [25] and [1]. McGovern suggests the use of a business based approach in order to avoid late design breakage and integration problems. A method similar to the requirements engineering approach with UMM is proposed in [5] using use case maps (UCMs). Such a scenario method allows the integration of different viewpoints e.g. business value viewpoint or business process viewpoint.

A Business Media Framework providing a solution for the integrated management of business transactions is presented in [35]. Next to the business requirements, the model presented also considers security and legal aspects in detail. A framework for service-oriented software engineering has been presented in [13]. The idea of the framework is to create a well defined business case in which the business processes and the business requirements are identified using UML and BPMN diagrams.

6. CONCLUSION AND OUTLOOK

In this paper we have introduced UN/CEFACT's Modeling Methodology (UMM) which we have co-edited. UMM is defined as a profile on top of UML 2.0 thus allowing an easy integration into a modeling tool of choice. We have demonstrated the different views and artifacts of UMM using an example from the waste management domain. The resulting UMM development process overcomes the limitations of UMM 1.0 as outlined in the introduction: First, the modeling of business documents is addressed by the concepts of the business information view. Second, business transactions may specify a set of alternative responses. Third, shared business entity states have been incorporated into business transactions. Each business transaction results in one of a set of well-defined business entity states. The resulting state depends on guards reflecting the type and/or structure of the response. Fourth, the guards of business collaboration protocols reflect the business entity states reached by the underlying business transactions and may thus be an appropriate source to derive machine processable process descriptions. Fifth, the business collaboration protocol enables the specification of nested collaborations which interlink activities of different business collaborations. Finally, the UMM has adapted an intuitive package structure grouping artifacts that naturally go together, e.g. each activity diagram is always a child of the use case describing its requirements.

Future work will concentrate on integrating value-based requirements engineering into the business requirements view. In value-based requirements engineering focus is laid on business models - being well distinguished from business process models - in order to survey the economic justification for e-business systems. Prominent approaches for analyzing business models are e3-Value methodology [4], the Resource-Event-Agent (REA) theory [3], or the business model ontology (BMO) [30]. We will investigate the integration of these methods as part of UMM's requirements engineering process. This permits the exploration of the economic rationale of a B2B system in early phases of the UMM development process.

We developed a UMM Add-In [10] on top of the UML modeling tool Enterprise Architect [38]. The Add-In is a tool support for UMM 1.0 and comprises a set of features such as model validation, semi-automatic generation of model artifacts, built in worksheet support the automatic derivation of deployment artifacts such as BPEL and WSDL. It is planned for the future to adapt the Add-In for UMM 2.0.

7. REFERENCES

[1] M. Erder and P. Pureur. Defining Business Requirements Quickly and Accurately. *IT Professional*, 6:51–56, 2004.

[2] E. Folmer and J. Bastiaans. Methods for Design of Semantic Message-Based B2B Interaction Standards. In *Enterprise Interoperability III*, pages 183–194. Springer London, April 2008.

[3] G. Geerts and W. E. McCarthy. The Ontological Foundation of REA Enterprise Information Systems. Technical report, Michigan State University, 2000.

[4] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce idea. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[5] J. Gordijn, H. de Bruin, and H. Akkermans. Scenario Methods for Viewpoint Integration in e-Business Requirements Engineering. In *Proceedings of the 34th Hawaii International Conference on System Sciences*. IEEE, 2001.

[6] B. Hofreiter and C. Huemer. Transforming UMM Business Collaboration Models to BPEL. In *Proceedings of OTM Workshops 2004*. Springer LNCS, 2004.

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

[8] B. Hofreiter, C. Huemer, P. Liegl, R. Schuster, and M. Zapletal. UN/CEFACT'S Modeling Methodology (UMM): A UML Profile for B2B e-Commerce. In *Advances in Conceptual Modeling - Theory and Practice, ER 2006 Workshops BP-UML*. Springer LNCS, 2006.

[9] B. Hofreiter, C. Huemer, P. Liegl, R. Schuster, and M. Zapletal. Deriving executable BPEL from UMM Business Transactions. In *Proceedings of the IEEE International Conference on Services Computing (SCC 2007)*. IEEE, 2007.

[10] B. Hofreiter, C. Huemer, P. Liegl, R. Schuster, and M. Zapletal. UMM Add-In: A UML Extension for UN/CEFACT's Modeling Methodology. In *Service-Oriented Computing - ICSOC 2007, Fifth International Conference*, volume 4749 of *Lecture Notes in Computer Science*, pages 618–619. Springer, 2007.

[11] 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 the Multikonferenz Wirtschaftsinformatik 2006, Passau Germany*, Lecture Notes in Informatics (LNI), pages 9–23, 2006.

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

[13] H. Karhunen, M. Jäntti, and A. Eerola. Service-Oriented Software Engineering (SOSE) Framework. In *Proceedings of 2005 International Conference on Services Systems and Services Management (ICSSSM)*. IEEE, 2005.

[14] H. Kim. Conceptual Modeling and Specification Generation for B2B Business Processes based on ebXML. *SIGMOD Rec.*, 31(1):137–145, 2002.

[15] B. Korherr and B. List. An Evaluation of Conceptual Business Process Modelling Languages,. In *21st ACM Symposium on Applied Computing (SAC'06)*. ACM Press, 2006.

[16] 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.

[17] R. M. Lee. Documentary Petri Nets: A Modeling Representation for Electronic Trade Procedures. In *Business Process Management: Models, Techniques, and Empirical Studies*. Springer, 2000.

[18] K. Lenz and A. Oberweis. Inter-organizational Business Process Management with XML Nets. In *Petri Net Technology for Communication-Based Systems*. Springer, 2003.

[19] F. Leymann, D. Roller, and M.-T. Schmidt. Web services and business process management. *IBM Systems Journal - New Developments in Web Services and E-commerce*, 41(2), 2002.

[20] P. Liegl, R. Schuster, and M. Zapletal. *UN/CEFACT's Modeling Methodology (UMM) 1.0 - A Guide to UMM and the UMM Add-In*. Verlag Dr. Müller, 2008.

[21] F.-R. Lin, M.-C. Yang, and Y.-H. Pai. A generic structure for Business Process Modeling. *Business Process Management Journal*, 8(1):19–41, 2002.

[22] S. Ling and S. W. Loke. Advanced Petri Nets for Modelling Mobile Agent Enabled InterorganizationalWorkflows. In *9th IEEE IntŠl Conf. and Workshop on the Engineering of Computer-Based Systems (ECBS 2002)*. IEEE Computer Society, 2002.

[23] B. List and B. Korherr. A UML 2 Profile for Business Process Modelling. In *ER 2005 Workshops Proceedings*, 2005.

[24] R. Mayer, C. Menzel, M. Painter, P. deWitte, T. Blinn, and B. Perakath. Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Technical report, Sept. 1995.

[25] F. McGovern. Managing software projects with business-based requirements. *IT Professional*, 4:18–23, 2002.

[26] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE, Vol.77, Iss.4*. IEEE Computer Society, 1989.

[27] OASIS Web Services Business Process Execution Language (WSBPEL) TC. *Web Services Business Process Execution Language Version 2.0*, Jan. 2007. Version 2.0, Commitee Specification.

[28] Object Management Group (OMG). *Business Process Modeling Notation*, 2006. Specification, Version 1.0.

[29] Object Management Group (OMG). *Unified Modeling Language Specification*, Nov. 2007. Version 2.1.2.

[30] A. Osterwalder and Y. Pigneur. An e-Business Model Ontology for Modeling e-Business. In *15th Bled Electronic Commerce Conference*, 2002.

[31] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-Oriented Computing: State of the Art and Research Challenges. *IEEE Computer*, 40(11):38–45, 2007.

[32] M. Penker and H.-E. Eriksson. *Business Modeling With UML: Business Patterns at Work*. Wiley, 2000.

[33] M. E. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, June 1998.

[34] C. Quyang, M. Dumas, A. H. ter Hofstede, and W. M. van der Aalst. From BPMN Process Models to BPEL Web Services. In *Proceedings 2006 IEEE International Converence on Web Services (ICWS'06)*, pages 285–292. IEEE, 2006.

[35] A. Rungeand, B. Schopp, and K. Stanoevska-Slabeva. The Management of Business Transactions through Electronic Contracts. In *Proceedings of the 10th Internaional Workshop on Database and Expert Systems Applications*. IEEE, 1999.

[36] N. Russell, W. M. van der Aalst, A. H. ter Hofstede, and P. Wohed. On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling. In *Third Asia-Pacific Conference on Conceptual Modelling (APCCM2006)*. Australian Computer Society, Inc., 2006.

[37] A.-W. Scheer. *ARIS - Business Process Modeling*. Springer, 2000.

[38] Sparx Systems. *Enterprise Architect*. Sparx Systems, May 2008.

[39] Supply Chain Council. *Supply-Chain Operations Reference-model - SCOR*, 2006. Version 8.0.

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

[41] UN/CEFACT. *UN/CEFACT's Modeling Methodology (UMM), UMM Meta Model - Foundation Module*, Mar. 2006. Technical Specification V1.0, http://www.unece.org/cefact/umm/UMM_Foundation_Module.pdf.

[42] UN/CEFACT International Trade and Business Process Group (TBG14). *UN/CEFACT Common Business Process Catalog*, Sept. 2005. Version 1.0, Technical Specification.

[43] UN/CEFACT TMG. *Core Components Technical*

Specification - Part 8 of the ebXML Framework. United Nations Center For Trade Facilitation and Electronic Business, 2003.

[44] UN/CEFACT TMG. *UML Profile for Core Components based on CCTS 2.01 (UPCC)*. United Nations Center For Trade Facilitation and Electronic Business, 2006.

[45] UN/CEFACT TMG. *UN/CEFACT Core Component Message Assembly*, Dec. 2007. First Working Draft.

[46] W. M. van der Aalst. Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets. *Systems Analysis - Modelling - Simulation*, 34(3):335–367, 1999.

[47] W. M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[48] W. M. P. van der Aalst. Making Work Flow: On the Application of Petri Nets to Business Process Management. In *Applications and Theory of Petri Nets 2002, 23rd International Conference, ICATPN 2002*. Springer LNCS, 2002.

[49] S. A. White. Using BPMN to Model a PBEL Process. *BPTrends*, 2005.

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