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].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

10th Int. Conf. on Electronic Commerce (ICEC) '08 Innsbruck, Austria Copyright 2008 ACM 978-1-60558-075-3/08 ...$5.00.
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 to demonstrate the adapted UMM 2.0 business process model 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.

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.

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 use case (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
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 discussed 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 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
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 define 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 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 (ReAction). 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 6: Overview of the Business Choreography View**

**Figure 7: Business Transaction Use Case**
Business transaction is defined in its own business transaction view and each included business collaboration is defined in its own business collaboration view.

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
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 importer 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-
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). The cross border consignment is connected to the information envelope using the concept of an association messaging business information entity (ASMBIE). 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 straightforward 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-technicians.

A more recent approach is the so called Business Process Modelling 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 breakdown 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


