Modeling e-Government processes with UMM

Philipp Liegl[1], Robert Mosser[2], Rainer Schuster[1] and Marco Zapletal[3]

[1]Research Studios Austria, Austrian Research Centers GmbH - ARC
[2]Institute for Distributed and Multimedia Systems, University of Vienna
[3]Institute of Software Technology, Vienna University of Technology
`pliegl@researchstudio.at, robert.mosser@univie.ac.at`
`rschuster@researchstudio.at, marco@ec.tuwien.ac.at`

Abstract

The United Nation's Center for Trade Facilitation and Electronic Business (UN/CEFACT)) is a standardization body known for its work on UN/EDIFACT and ebXML. One of its most recent developments is UN/CEFACT's Modeling Methodology (UMM). The UMM standard is used to model interorganizational business processes in the B2B domain. With the increasing presence of governmental institutions and their services on the internet, the frontier between B2B and B2G/G2G vanishes. Today one expects a governmental institution to react like any other business partner. Therefore also governments now face the interoperability and compatibility issues as regular businesses do. In order to allow two governmental institutions to collaborate, a methodology uniquely depicting the inter-institutional process from a global perspective is needed. In this paper we propose to use UN/CEFACT's Modeling Methodology in the eGovernment domain. UMM allows the definition of a global choreography which is then being used to derive local choreographies for each business partner. Exemplarily a real-world scenario from the waste transport domain within the European Union will be shown. Furthermore the possible integration in the context of the We-Go project[1] will be examined.

1 Introduction

The multitude of e-Government initiatives and efforts which have been made in the last few years have shown the significance of the e-Government domain. With more and more governmental institutions being present on the internet, other businesses expect a governmental institution to react like a regular business does e.g. offering key business functionalities via services accessible over the internet. When fulfilling these expectations governmental institutions face the same problems as regular businesses. Therefore the distinction commonly made between government-to-government (G2G) and business-to-business (B2B) and business-to-government (B2G) is not appropriate any more. In order to help governmental institutions to fulfill the customer needs we propose to use a methodology for modeling the processes in the eGovernment domain. The United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT), known for its standardization work in the field of UN/EDIFACT [18] and ebXML [11], has developed the so called UMM standard. UMM stands for UN/CEFACT's Modeling Methodology and enables to capture business requirements independent of the underlying implementation technology such as Web Services or ebXML. With the help of UMM a global choreography can be defined which forms the basis for deriving the local choreographies for each business partner.

The remainder of this paper is structured as follows: Section 2 explains the UMM standard in detail by using an example from the waste transport domain. In section 3 we introduce our UMM modeling tool called UMM Add-In and its support features for the UMM standard. The integration of the UMM standard in the roll-out of the We-Go project will be the subject of section 4. Section 5 gives an overview about other research efforts in the field of inter-organizational business processes. The paper concludes with an outlook and future research issues.

[1]http://www.wego-project.eu Proposal No 045472 Contract No 045472

2 UN/CEFACT's modeling methodology

UN/CEFACT's Modeling Methodology (UMM) is a UML based methodology for capturing the requirements in an inter-organizational business process. It is independent of the underlying transfer syntax. The overall goal of the UMM methodology is to create a global choreography of the business process. If two business partners interacting with each other each defined their own choreography for the business process the resulting choreographies are unlikely to match. UMM pursues a top down approach by first defining the global choreography from which the local choreographies are derived. Hence it is ensured, that both choreographies are complementary.

UMM is built upon the UML meta model and defined as a UML profile [20] e.g. a set of stereotypes, tagged values and OCL constraints. Its current version is 1.0 based on UML 1.4 [12]. The predecessor based on UML 2.0 is currently under development.

In the following section we will outline a real-world UMM example from the eGovernment domain in the European Union.

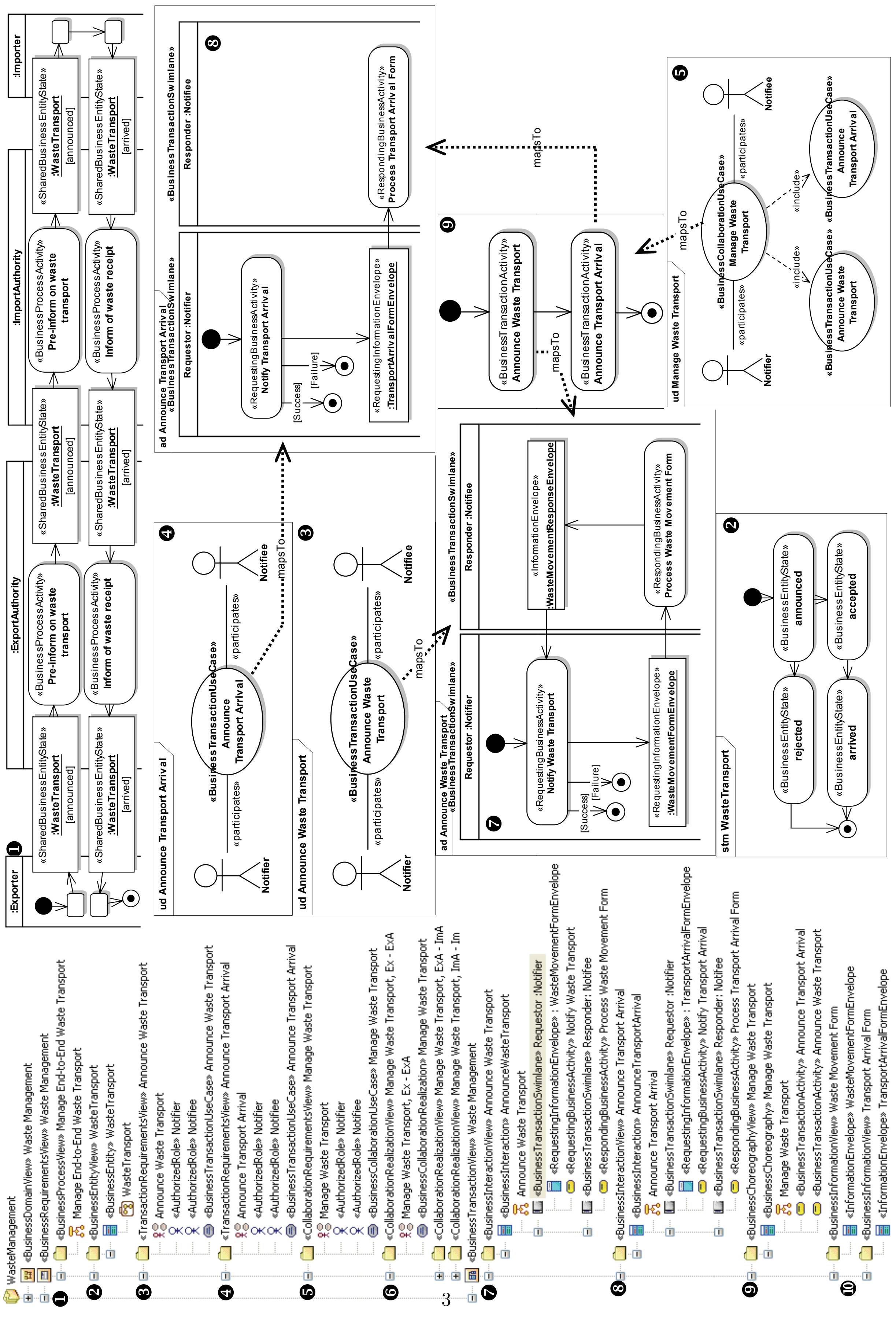
2.1 A UMM example from the e-Government domain

In this section we briefly describe the steps of UMM and the resulting artifacts. For a better understanding we walk through the UMM by the means of a simple example of the waste transport domain as it is used in the European Union today. A waste transport taking place between two countries in the European Union will be analyzed. The transport of goods and persons within the EU is not subject to strict regulations any more. However the transport of waste between two European countries underlies regulations accompanied by a multitude of forms and administrative documents. One major goal of the European Union is to facilitate the waste transport by supporting the transport process with information technology means and to decrease the amount of paper documents involved.

The relevant artifacts of our example are depicted in figure 1. On the left hand side of this figure we see the structure of our *waste management* model. A UMM *business collaboration model* comprises three main views: the *business domain view* (BDV), the *business requirements view* (BRV), and the *business transaction view* (BTV). The three top level packages of any UMM model are always stereotyped in accordance to these views. For relevant artifacts in the tree-view on the left hand side of figure 1 the according diagrams are shown on the right hand side of the figure.

The BDV is used to gather existing knowledge from stakeholders and business domain experts. In interviews the business process analyst tries to get a basic understanding of the business processes in the domain. The use case descriptions of a business process are on a rather high level. One or more *business partner types* participate in a business process and zero or more stakeholders have an *interest in* dependency with the process. The BDV results in a map of business processes, i.e. the business processes are classified. Thus, the BDV package includes *business area* sub-packages. UN/CEFACT suggests to use *business areas* according to the Common Business Process Catalog [19]. Each *business area* consists of *process area* packages that correspond to the Open-edi phases (planning, identification, negotiation, actualization, and post-actualization) [5]. In our *waste management* example the relevant *business areas* are *Logistics* and *regulation*. *Logistics* covers at least the *process areas* of *actualization* and *post-actualization*. The *business area regulation* only contains the *process area actualization*. We do not want to detail here all the business processes that may be important to the domain experts and stakeholders in these areas. Due to size restrictions we do not present the structure of *business areas* and *process areas* in the *business domain view* of figure 1.

The BRV consists of a number of different sub-views. The *business process view* (1) and the *business entity view* (2) are both very project specific. The *business process view* gives an overview about the business processes, their activities and resulting effects, and the business partners executing them. The activity graph of a business process may describe a single partners process, but may also detail a multi-party choreography. The business process analyst tries to discover interface tasks creating/changing business entities that are shared between business partners and thus, require communication with a business partner. Discovery of interface task is more important in this step than modeling an exact control flow of activities. In our example we detail a multi-party business process for an *end-to-end waste transport* (1). The four parties involved in the process are *exporter*, *export authority*, *import authority* and *importer*. Due to space limitations the *exporter* and the *importer* have been simplified. The *exporter* pre-informs the *export authority* about an upcoming *waste transport*. The *export authority* in turn informs the *import authority* of the country the *waste transport* is imported to. The *import authority* then informs the *importer*. After the *waste transport*



WasteManagement
«BusinessDomainView» Waste Management
«BusinessRequirementsView» Waste Management
«BusinessProcessView» Manage End-to-End Waste Transport
Manage End-to-End Waste Transport
«BusinessEntityView» WasteTransport
«BusinessEntity» WasteTransport
WasteTransport
«TransactionRequirementsView» Announce Waste Transport
Announce Waste Transport
«AuthorizedRole» Notifier
«AuthorizedRole» Notifiee
«BusinessTransactionUseCase» Announce Waste Transport
«TransactionRequirementsView» Announce Transport Arrival
Announce Transport Arrival
«AuthorizedRole» Notifier
«AuthorizedRole» Notifiee
«BusinessTransactionUseCase» Announce Transport Arrival
«CollaborationRequirementsView» Manage Waste Transport
Manage Waste Transport
«AuthorizedRole» Notifier
«AuthorizedRole» Notifiee
«BusinessCollaborationUseCase» Manage Waste Transport
«CollaborationRealizationView» Manage Waste Transport, Ex - ExA
Manage Waste Transport, Ex - ExA
«BusinessCollaborationRealization» Manage Waste Transport
«CollaborationRealizationView» Manage Waste Transport, ExA - ImA
«CollaborationRealizationView» Manage Waste Transport, ImA - Im
«BusinessTransactionView» Waste Management
«BusinessInteractionView» Announce Waste Transport
«BusinessInteraction» AnnounceWasteTransport
Announce Waste Transport
«BusinessTransactionSwimlane» Requestor :Notifier
«RequestingInformationEnvelope» : WasteMovementFormEnvelope
«RequestingBusinessActivity» Notify Waste Transport
«BusinessTransactionSwimlane» Responder: Notifiee
«RespondingBusinessActivity» Process Waste Movement Form
«BusinessInteractionView» Announce Transport Arrival
«BusinessInteraction» AnnounceTransportArrival
Announce Transport Arrival
«BusinessTransactionSwimlane» Requestor :Notifier
«RequestingInformationEnvelope» : TransportArrivalFormEnvelope
«RequestingBusinessActivity» Notify Transport Arrival
«BusinessTransactionSwimlane» Responder: Notifiee
«RespondingBusinessActivity» Process Transport Arrival Form
«BusinessChoreographyView» Manage Waste Transport
«BusinessChoreography» Manage Waste Transport
Manage Waste Transport
«BusinessTransactionActivity» Announce Transport Arrival
«BusinessTransactionActivity» Announce Waste Transport
«BusinessInformationView» Waste Movement Form
«InformationEnvelope» WasteMovementFormEnvelope
«BusinessInformationView» Transport Arrival Form
«InformationEnvelope» TransportArrivalFormEnvelope
1 2 3 4 5 6 7 8 9 10
:Exporter
:ExportAuthority
:ImportAuthority
:Importer
«SharedBusinessEntityState» :WasteTransport [announced]
«SharedBusinessEntityState» :WasteTransport [arrived]
«BusinessProcessActivity» Pre-inform on waste transport
«BusinessProcessActivity» Inform of waste receipt
ud Announce Transport Arrival
«BusinessTransactionUseCase» Announce Transport Arrival
«participates»
Notifier
Notifiee
mapsTo
ud Announce Waste Transport
«BusinessTransactionUseCase» Announce Waste Transport
ad Announce Transport Arrival
«BusinessTransactionSwimlane» Requestor :Notifier
«BusinessTransactionSwimlane» Responder :Notifiee
«RequestingBusinessActivity» Notify Transport Arrival
[Success]
[Failure]
«RequestingInformationEnvelope» :TransportArrivalFormEnvelope
«RespondingBusinessActivity» Process Transport Arrival Form
ad Announce Waste Transport
«RequestingBusinessActivity» Notify Waste Transport
«RequestingInformationEnvelope» :WasteMovementFormEnvelope
«InformationEnvelope» :WasteMovementResponseEnvelope
«RespondingBusinessActivity» Process Waste Movement Form
«BusinessTransactionActivity» Announce Waste Transport
«BusinessTransactionActivity» Announce Transport Arrival
ud Manage Waste Transport
«BusinessCollaborationUseCase» Manage Waste Transport
«include»
«BusinessTransactionUseCase» Announce Waste Transport
«BusinessTransactionUseCase» Announce Transport Arrival
stm WasteTransport
«BusinessEntityState» announced
«BusinessEntityState» accepted
«BusinessEntityState» rejected
«BusinessEntityState» arrived

Figure 1. UMM Overview

has been conducted the *importer* informs the *import authority* about the arrived *waste transport*. The *import authority* then informs the *export authority* of the country the *waste transport* originated from. Finally the *export authority* notifies the *exporter* about the successful *waste transport*.

The information exchanged between business partners is about the business entities *waste transport*. A *waste transport* object is created with the state *announced*. Later it is set either to the state *rejected* or *accepted*. Finally, after the waste transport took place the *waste transport* state is set to *arrived*. These so-called *shared business entity* states must be in accordance with the *business entity lifecycle* of *waste transport*. The lifecycle is defined in the state chart of the *business entity view* (2). Due to space limitations the process view in (1) has been simplified and the options for setting the *waste transport* to *accepted* or *rejected* have been left out.

Since the same tasks take place between a different pair of *business partner types*, it is not appropriate to describe these tasks for each pair again and again. Instead, these tasks are defined between *authorized roles*. A *transaction requirements view* defines the *business transaction use case* for a certain task and binds the two *authorized roles* involved. The *authorized roles* are defined in the exact context of the *business transaction use case*. In our example we have two *transaction requirements views*: *announce waste transport* (3), and *announce transport arrival* (4). By coincidence the *authorized roles* in *announce waste transport* and *announce transport arrival* have the same name, namely *notifier* and *notifiee*.

The *collaboration requirements view* includes a *business collaboration use case*. The *business collaboration use case* aggregates *business transaction use cases* or nested *business collaboration use cases*. This is manifested by *include* associations. In our example the *business collaboration use case manage waste transport* (5) includes the *business transaction use cases announce waste transport* (3) and *announce transport arrival* (4). Furthermore, the *authorized roles* participating in the *business collaboration use case* must be defined within the context and namespace of the *collaboration requirements view*. We call the roles *notifier* and *notifiee*. Again the homonymous names are coincidental. The *notifier* is the one who initiates the *manage waste transport* and the *notifiee* is the one who reacts on it. The *authorized role notifier* in *announce waste transport arrival* (4) is not the same as the one in *manage waste transport* (5). This means, we have two *authorized roles notifier*, each defined in the namespace of its view. *Maps to* dependencies are used to define which *authorized role* of a *business collaboration use case* plays which role in an *included business transaction use case* (or nested *business collaboration use case*). In our example the *notifer* of *manage waste transport* (5) plays the *notifier* of *announce waste transport* (3) but also the *notifier* in *announce transport arrival* (4). The *notifiee* of *manage waste transport* (5) plays the *notifiee* of *announce waste transport* (3), but also the *notifiee* of *announce transport arrival* (4).

A *business collaboration use case* may have many *business collaboration realizations* that define which *business partners* play which *authorized roles*. Due to space limitations the concept of *business collaboration realizations* is not further elaborated here.

The BTV builds upon the BRV and defines a global choreography of information exchanges and the document structure of these exchanges. The choreography described in the requirements of a *business transaction use case* is represented in exactly one activity graph of a *business transaction*. A *maps to* dependency between them allows traceability between the requirements and the *business transaction*, which is defined in a *business interaction view*. In our example, the *announce transport arrival* requirements (4) are mapped to a corresponding choreography (8). The same mapping is made for the *announce waste transport* requirements (3+7).

A *business transaction* is characterized as follows: If an *authorized role* recognizes an event that changes the state of a *business entity*, it initiates a *business transaction* to synchronize with the collaborating *authorized role*. A *business transaction* is an atomic unit that leads to a synchronized state in both information systems. We distinguish one-way and two-way *business transactions*: In the former case, the initiating *authorized role* reports an already effective and irreversible state change that the reacting *authorized role* has to accept. In the other case, the initiating partner sets the *business entity/ies* into an interim state and the final state is decided by the reacting *authorized role*. It is a two-way transaction, because *business information* flows from the initiator to the responder to set the interim state and backwards to set the final and irreversible state change. Irreversible means that returning to an original state requires compensation by another *business transaction*.

Owing to this strict definition, a UMM *business transaction* follows always the same pattern: A *business transaction* is performed between two *authorized roles* that are already known from the *business transaction use case* and that are assigned to exactly one *business transaction swimlane* each. Each *authorized role* performs exactly one activity. An *object flow* between the *requesting* and the *responding business activity* is mandatory. An *object flow* in the reverse direction is optional. In our example the *business transactions* announce waste transport (7) is a two-way transaction whereas announce transport arrival (8) is a one-way transaction. Sending the *waste movement form envelope* (7) sets the interim state *arrived* of the *business entity waste transport*. The reply in the *announce waste transport* sets it to the state *accepted* or *rejected*. Finally the notification in *announce transport arrival* (8) sets the final state of the *business entity waste transport* to *arrived*.

The requirements described in a *business collaboration use case* are choreographed in the activity graph of a *business collaboration protocol*, which is defined in a *business choreography view*. This one-to-one relationship is denoted by another *maps to* dependency. In our example, the manage waste transport requirements (5) are mapped to the *business collaboration protocol* shown in (9). A *business collaboration protocol* choreographs a set of *business transaction activities* and/or *business collaboration activities*. A *business transaction activity* is refined by the activity graph of a *business transaction*. In our example, the *business collaboration protocol* of manage waste transport (9) consists of two *business transaction activities*: announce waste transport and announce transport arrival. Each of them is refined by its own *business transaction* (7,8). *Maps to* dependencies keep track of this refinement. *Business collaboration activities* - which are not used in our example - are refined by a nested *business collaboration protocol*.

The *business information views* are used to define the structure of *business documents* exchanged in *business transactions*. Each of the four *information envelopes* exchanged in the two *business transactions* lead to a *business information view* describing the envelope's document structure. Due to space limitations only two *information envelopes* are shown namely waste movement form envelope and transport arrival form envelope. An *information envelope* consists of a *header* and a *body*. The *header* carries auxiliary information and the *body* holds the actual business document exchanged during the transaction. UMM itself does not prescribe a particular format for the exchanged business document. However the use of so called core components [21][23] is suggested.

3 A tool support for UMM

Since UMM is defined as a UML profile, a business analyst may use any UML tool to model UMM business collaboration models. As we outlined in the beginning, UMM artifacts are based on a specific subset of UML to capture complex business collaborations. Between these artifacts, a number of dependencies and constraints exist. If a regular UML tool is used for UMM, these rules are not enforced. Furthermore, a modeler is not prevented from using modeling elements that are not part of UMM's meta model. However, valid models are required for at least two reasons. Firstly, if a business process model is shared between partners, it has to be formally correct in order to ensure an unambiguous communication in terms of the modeled business domain. Secondly, a UMM model allows the generation of machine-interpretable process and business document specifications to configure B2B information systems. Obviously, deriving artifacts following a model driven approach works only for valid models. Consequently, a modeling tool considering UMM-specifics is needed. Instead of developing a new modeling tool from scratch, we decided to build our UMM tool on top of the commercial UML tool Enterprise Architect. The UMM Add-In utilizes Enterprise Architect's UML functionality and extends it by implementing the UMM-specifics. The UMM Add-in supports the business analyst by the following main features:

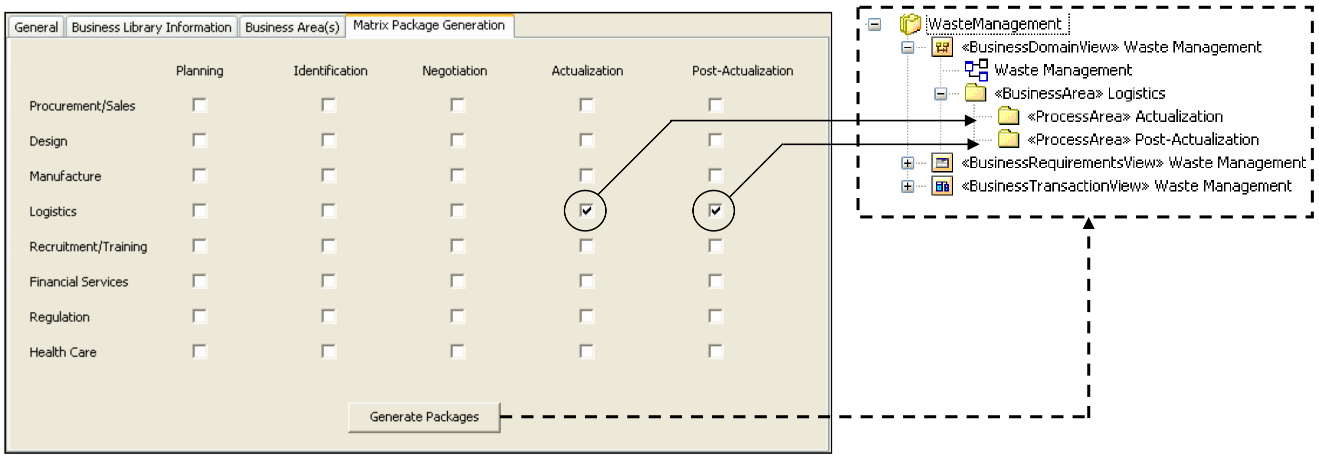
1. UMM-specific toolbar
2. UMM requirements engineering support
3. Semi-automatic generation of UMM artifacts
4. Validation of the UMM model
5. Generating process specifications for B2B information systems
6. Core Component support

3.1 UMM specific toolbar

In order to create a UMM model it is convenient to drag and drop UMM stereotypes from a toolbar onto the modeling canvas. Thus, the stereotypes as defined in the UML profile for UMM are integrated into Enterprise Architect and provided in a toolbar. The toolbar itself is organized in sections that correspond to the UMM views. This helps the user applying the right stereotypes in a certain UMM step. In order to prevent the user from using UML elements not part of the UMM subset, the toolbar is restricted to the UMM stereotypes and UML standard elements required by UMM.

3.2 UMM requirements engineering support

Requirements engineering is not only important for software development processes, but is also an criti-


General
Business Library Information
Business Area(s)
Matrix Package Generation
Generate Packages
WasteManagement
«BusinessDomainView» Waste Management
Waste Management
«BusinessArea» Logistics
«ProcessArea» Actualization
«ProcessArea» Post-Actualization
«BusinessRequirementsView» Waste Management
«BusinessTransactionView» Waste Management

	Planning	Identification	Negotiation	Actualization	Post-Actualization
Procurement/Sales	☐	☐	☐	☐	☐
Design	☐	☐	☐	☐	☐
Manufacture	☐	☐	☐	☐	☐
Logistics	☐	☐	☐	✓	✓
Recruitment/Training	☐	☐	☐	☐	☐
Financial Services	☐	☐	☐	☐	☐
Regulation	☐	☐	☐	☐	☐
Health Care	☐	☐	☐	☐	☐

Figure 2. Semi-automatic generation of BDV's initial package structure

cal task in business process modeling. Usually a lot of different people with different background are involved in the development process of a business model. In order to close this gap of various levels of business knowledge and expertise, the requirements of a business model must be documented in natural language. The outcome of this process is a common understanding of each part of the business scenario. Usually the business knowledge is collected during interviews between the business analysts and the business domain experts.

Especially in UMM the requirements engineering plays a major role. UN/CEFACT recommends using a set of so-called worksheets in order to capture the gathered business information. A UMM worksheet has its pre-defined structure containing standardized elements and is assigned to a specific part in the business model. Traditionally, worksheet templates have been filled out by using a word processor. This leads to a great number of external paper documents. Since most of the worksheet information is captured later on in tagged values, this result in duplication of efforts and a danger of inconsistency. Thus the worksheet editor of the UMM Add-In integrates the requirements engineering into the UMM model. Once the business domain knowledge is acquired, the information is directly stored in the corresponding tagged value of the assigned UMM artifact.

The worksheet editor offers an interactive form including all elements specified by UN/CEFACT. Additionally self-defined worksheets may be created using an XML based worksheet definition language. Each UMM artifact, which needs to be documented, has an XML based property file, in order to specify the layout, the structure and the content of the worksheet. This guarantees a flexible adaptation of worksheets to special business needs and to changes due to an update of the UMM meta model. In order to ensure traceability, some artifacts of a UMM model are interlinked - e.g. the *business transaction* in the *business transaction view* has a dependency to the corresponding *business transaction use case* in the *business requirements view*. Thus, business requirements information may not differ dramatically between such UMM worksheets. In order to avoid inconsistency the worksheet editor provides the modeler with business information already entered to prior worksheets having semantically the same content.

The communication between the business domain expert and the business analysts is usually based on some model reports. Thus, such reports must be generated from the model. The modeler may generate document reports using the export functionality of the worksheet editor. The user either exports the gathered information into Microsoft Word, or publishes HTML pages of the business model requirements. Furthermore, the whole documentation can be stored in an XML based format for backup purposes.

3.3 Semi automatic generation of UMM artifacts

In the section above we have demonstrated, how the worksheet editor is used for capturing business requirements of a UMM model. The worksheet editor offers an added value by generating UMM artifacts automatically. Thereby, the modeler is released from some routine tasks. Instead of modeling recurring UMM patterns manually the UMM Add-In creates such artifacts automatically using the input of the worksheets the modeler already entered.

In figure 2 such a workflow is depicted. The worksheet of the *business domain view* captures information about *business areas* and *process areas.* Since UMM recommends using a set of hierarchically predefined *business* and *process areas* according to the Common Business Process Catalog (CBPC) [19], the package structure of the BDV differs only by the modeler's choice of package constellation. The structure of *business areas* and *process areas* - which may follow a matrix - is transformed into a tree-structure of packages. Instead of manually creating this tree-structure, the modeler simply checks the relevant cells in the predefined matrix and the package structure is created. In our *waste management* example the *business area Logistics* together with its *process areas actualization* and *post-actualization* have been generated automatically using this matrix.

Another candidate for a semi-automatic generation is the *business transaction.* The structure of a *business transaction* follows always the same pattern. However, instances of *business transactions* differ from each other with respect to the exchanged information, the names of the activities and the participating roles. The *business transaction announce waste transport* has been generated by the UMM Add-In using the form depicted in figure 3. Since every *business transaction* has a corresponding *business transaction use case*, the UMM Add-In offers the modeler a list of all *business transaction use cases* already defined in the BRV. Once a use case is selected, the modeler is presented with possible *participating roles* and *information envelopes.* Thereby the modeler saves a lot of time instead of manually creating each *business transaction* from the scratch.

3.4 Validation of a UMM model

With the increasing complexity of a UMM model an inexperienced user often loses the overview and possible flaws in the model remain undetected. A foible in the model such as a missing connector might not be an issue if the model is just used to communicate the structure of the inter-organizational business process between humans. If, however, the model is used to derive business artifacts in an automated manner, a syntactically valid model is a prerequisite. As shown in the next chapter the UMM model can for instance be used to generate BPEL or BPSS documents. If the model is not correct the generator will most likely produce invalid documents or the generation fails completely.

Given these prerequisites, the UMM Add-In has a validation feature that allows checking the formal validity of a given UMM model. The validator is based on the OCL constraints incorporated in the UMM foundation module [20]. It, however, does not interpret the OCL constraints directly but the logic is incorporated in the program code. The validator feature of the UMM Add-In allows two types of validation runs - bottom-up validation and top-down validation. The former one is used to validate sub-packages of a given UMM model and the latter one is used to perform a validation run on the whole model. After a validation run the errors found in the model are presented to the modeler. By double-clicking on an error message a detailed description of the error is shown.

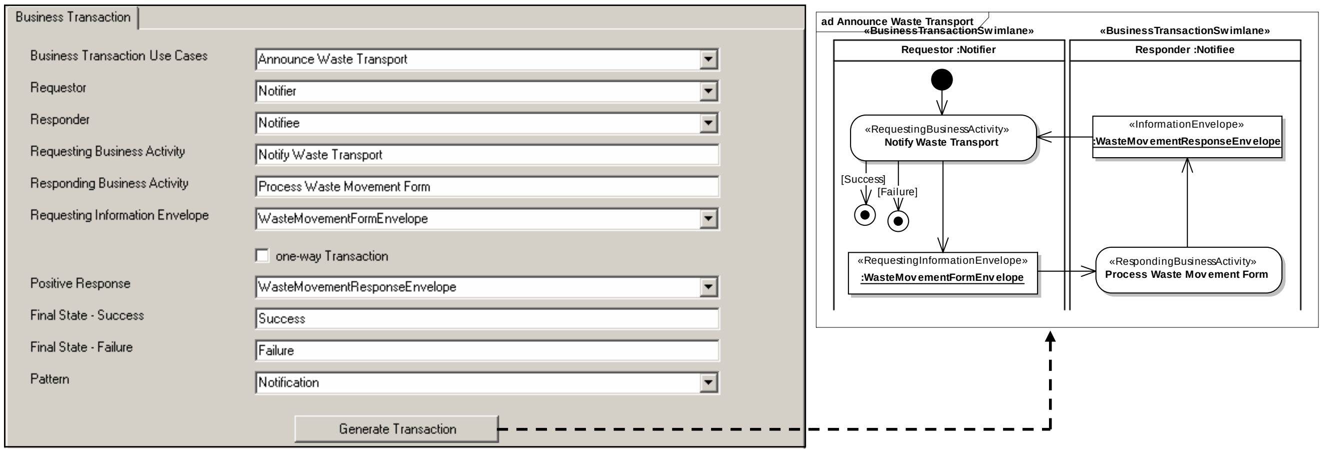
Apart from the possibility to validate a given UMM model the validator will also help new users to get acquainted to the UMM standard. Many users don't read the whole specification but start modeling from scratch on a try and error basis. Especially the bottom up validator will gradually help the inexperienced user to create a valid UMM model and antagonize the argument, that UMM is too hard to learn or apply.

3.5 Generating process specifications for B2B information systems

In our UMM Add-In, we support the derivation of process descriptions for Web Services and ebXML - the two major approaches for implementing SOAs. In case of Web Services, we support the Business Process Execution Language (BPEL), for ebXML the tool generates choreographies according to the Business Process Specification Schema (BPSS). On a conceptual level, mappings from UMM to BPSS [4] and from UMM to BPEL [3] have already been proposed. In our tool, the implemented transformation algorithms follow these approaches.

3.5.1 Generating BPEL from UMM.

BPEL describes the flow of a business process as a sequence of interactions between Web Services. An interaction is represented by an activity pointing to the respective service. Business partners participating in a process might provide services to the process and consume other partner's services. BPEL denotes the sequence of such Web Service calls, maps service calls to concrete Web Services and collates service consuming and service providing to the process participants. Thereby, BPEL describes the respective business process from a specific partner's point of view. Considering a business process internal to a company, whereby the company fully controls the logic of the process, the BPEL approach works fine. The process can easily be composed by orchestrating the required services in


Business Transaction

Business Transaction Use Cases	Announce Waste Transport
Requestor	Notifier
Responder	Notifiee
Requesting Business Activity	Notify Waste Transport
Responding Business Activity	Process Waste Movement Form
Requesting Information Envelope	WasteMovementFormEnvelope
	☐ one-way Transaction
Positive Response	WasteMovementResponseEnvelope
Final State - Success	Success
Final State - Failure	Failure
Pattern	Notification

Generate Transaction
ad Announce Waste Transport
«BusinessTransactionSwimlane»
«BusinessTransactionSwimlane»
Requestor :Notifier
Responder :Notifiee
«RequestingBusinessActivity»
Notify Waste Transport
«InformationEnvelope»
:WasteMovementResponseEnvelope
[Success]
[Failure]
«RequestingInformationEnvelope»
:WasteMovementFormEnvelope
«RespondingBusinessActivity»
Process Waste Movement Form

Figure 3. Semi-automatic generation of a business transaction

order to achieve the desired output. However, considering a complex B2B scenario that requires the participants to agree on process choreography, an approach whereby each participant describes its own view on the process in isolation will fail. In other words, if each business partner of a collaborative process describes its own BPEL the resulting process descriptions characterizing the same process will most likely not match. Thus, having BPEL applied in a B2B environment necessitates first a global view on the process, like UMM provides it, in order to gain complementary process descriptions.

In order to support this approach, the UMM Add-In allows the generation of partner-specific BPEL processes from a global UMM choreography. Since services are referenced by their WSDL in BPEL, the tool generates also a WSDL for each partner, describing the set of services he or she has to provide. Due to space limitations, we do not show the actual BPEL result.

3.5.2 Generating BPSS from UMM.

BPSS denotes the choreography of a B2B business process for ebXML environments. It captures the flow of a collaborative process from a global viewpoint describing how each participant has to act in order to fulfilf his or her part of the business process. BPSS is defined as an executable subset of the UMM specifying a B2B process by the concepts of a business collaboration consisting of business transactions. Although the BPSS specification does not mandate using UMM to gain BPSS process definitions, its use is recommended. Due to its alignment to UMM, BPSS allows the configuration of a business service interface according to the rich semantics of UMM business collaboration models. Given the space limitations the BPSS output is not show either.

4 UMM in the WeGo project context

The WeGo project *Enhancing Western Balkan eGovernment Expertise* targets to boost eGovernment awareness and knowledge in order for Western Balkan Countries (WBC) to reach higher productivity and equity. Therefore the project aims to transfer and adopt successful eGovernment best practices and knowledge. The WeGo project will raise the awareness of interoperability issues and cooperation as well as EU conformance. One of the main objectives of the project is the establishment of the WeGo Interoperability Framework with focus on transactional cross-border services and EU alignment. In addition, best practices for corresponding transactional application domains will be delivered as demonstration prototypes. Services addressed by the project include eCustoms - New Computerised Transit System (NCTS), eJustice cross-border cases - Automation of Court Procedures (ACP), European Companies Register (ECR) and European Land Information System (EULIS), eAdministration - Electronic Filing System (implementation of paperless government) and eTrade Facilitation for European Waste Transport (EUDIN). Another goal is the establishment of eGovernment Academies to support and complement the Interoperability Framework and demonstration prototypes efforts and to establish a solid foundation for future regional eGovernment course program deployments. Finally the project aims on the establishment of an eGovernment Knowledge Net built by a system of federated registries to give easy access to miscellaneous pieces of information, such as specifications or complete services including the solution and documentation. The WeGo Knowledge Net will encourage the re-use of existing approaches that will increase the likelihood of interoperability out of the box.

The analysis phase for the WeGo Interoperability Framework showed that there is a high need for clear and structured modeling and description of processes in the eGovernment area. Governments face interoperability issues when establishing cross-organizational services between different public authority bodies within the same or even different countries or services including European Union institutions. As mentioned above, cross-national services are covered by the WeGo project, e.g. in the eJustice and eCustoms area. The success of eGovernment heavily depends on the appropriate design and specification of processes in the application domain. Although the requirements in the various eGovernment domains may vary (as clearly indicated by the WeGo project) the specification of eGovernment processes should follow general, well-accepted design principles, such as UMM. A uniform structure and modeling of inter-organizational processes helps to increase business process interoperability. UMM is qualified to support that effort since it is a methodology for unambiguous definition of inter-organizational business processes. As an example, the eTrade Facilitation for European Waste Transport (EUDIN) system is not only based on IT-platform standards such as XML and Web Services, but also on standards describing the business logic of the system. One of the latter is UMM that has been used to define the business processes.

UMM is powerful but can be a challenge for users not familiar with it. Knowledge required to make use of UMM can be made available through the WeGo Knowledge Net by providing specifications, tutorials or complete business process models for re-use. Another channel for the distribution of UMM know-how are the WeGo Academies.

5 Related Work

Over the last couple of years, a lot of methodologies for modeling business processes have been developed. Some of them are based on special notations often defined by standardization bodies. Others customize the UML for business process modeling needs. Traditionally, business process modeling focuses on modeling business processes internal to an organization fulfilling customer needs. More recent approaches also take inter-organizational business processes into account. Another criteria for distinguishing business process modeling approaches is its binding to the supporting IT infrastructure. Some approaches are mainly used in the requirements specification phase to support the communication with business domain experts. Resulting models usually hide implementation complexity and are on a rather abstract level. Other approaches are more implementation oriented and rather provide a graphical interface for workflow languages or Web Service orchestrations/choreographies.

Recently, the Business Process Modeling Notation (BPMN) [13] has attracted a lot of attention. BPMN has been developed by the Object Management Group (OMG) in order to enhance a standardized modeling notation, which is readily understandable by all business users - from the business analysts to the technical developers. In order to realize this goal, BPMN incorporates aspects of already advanced modeling notations (e.g. UML activity diagrams, IDEF [10], ebXML BPSS [22], RosettaNet [15], etc.). In order to close the gap between the business process design and the business process implementation, OMG's Business Process Management Initiative (BPMI) standardizes the mapping from BPMN to an XML based executable business process, such as BPEL (Business Process Execution Language) [1].

Another very popular notation in business engineering are the Event-driven Process Chains (EPCs). EPC is a business process modeling language, focusing on control flow dependencies of activities in a business process. It is utilized in the ARchitecture of Integrated Information Systems (ARIS) by Scheer [16] as the central method for the conceptual integration of the functional, organizational, data, and output perspective in information systems design.

In addition to special business modeling notations, some approaches extend the UML meta model for this purpose. Initially, UML was designed to capture requirements for object oriented software systems in a graphical manner. Nevertheless some UML concepts, like UML activity diagrams, may be adopted for business process modeling. Korherr and List propose a UML profile for modeling business processes considering business process goals and performance measures [7]. Since the UML 2 activity diagram does not include such concepts, the UML meta model has been extended by a set of stereotypes and tagged values. On the one hand side this approach focuses on integrating measurable values into a business model at the conceptual layer. On the other hand side it specifies a mapping of UML models to BPEL, in order to make these values available for execution and monitoring. However, this approach is developed for modeling business processes internal to an organization. Further UML based approaches restricted to an internal view are proposed in

[9] and [14].

For the purpose of representing and managing B2B business processes considering an inter-organizational view, Kim proposes a UML 1.x based modeling approach [6]. He uses activity diagrams for modeling collaborative processes as a flow of transactions in order to create an ebXML compliant business process specification. A transaction is a message exchange between two business partners and is represented as an activity in the activity graph. The flow of data exchanged between business partners is then captured in sequence diagrams.

A more IT-oriented approach is proposed by Tyndale-Biscoe et al. [17]. The authors present a UML 1.4 profile for the integration of business processes and software development used in an EU funded project. The UML profile provides the mapping between real world business concepts and software artifacts.

Kramler et al. [8] use UML 2 for depicting the choreography of Web Services. Unlike traditional business process modeling, additional requirements must be considered for service collaborations - e.g. security management or transaction management. In their paper the authors split the models into a layered architecture - collaboration, transaction, and interaction level, in order to compare these levels of granularity with the eCo framework [2]. The modeling technique is based on the considerations for a mapping to the Business Process Specifications Scheme (BPSS) and the Business Process Execution Language (BPEL).

5.1 Conclusion and Outlook

In this paper we have presented UN/CEFACT's Modeling Methodology (UMM), a language for designing B2B processes. UMM is defined as a profile for UML 1.4. In other words, the UML meta model has been extended by a set of stereotypes, tagged values and constraints in order to adjust UML for the semantics required by B2B. In theory, UMM models can be created by any UML tool. However, as we outlined there is a significant need for a customized UMM tool considering the UMM-specifics. Therefore, we developed the UMM Add-In - a plug-in for the commercial UML tool Enterprise Architect. This tool supports business analysts in creating UMM compliant business collaboration models. The functionality of the Add-In was characterized in the main part of this paper.

UN/CEFACT's Modeling Methodology in combination with our UMM Add-In is already in use in a couple of - mostly e-Government related - projects for capturing collaborative business processes. The German Government uses UMM in order to define business collaborations between the local and federal governments. In this project a so-called XÖV Framework is responsible for a harmonization of different XML based standards used in the public administration by using the concepts of UMM and Core Components. A further project using UMM has been set up by the Australian Government and is called GovDex. The Department of Finance and Administration uses UMM and Core Components in order to improve the efficiency and effectiveness of business process collaborations across policy portfolios (e.g., taxation, human services, etc.) and administrative jurisdictions. The Canadian Administration is trying to apply the principles of UMM to describe the activities of governmental agencies. This work is known as the Governments of Canada Strategic Reference Model (GSRM).

Future research efforts will concentrate on the development of the successor of the current UMM standard - UMM 2.0. At the time when UMM 1.0 was developed the support for UML 2.0 in modeling tools was poor. However today UML 2.0 is state of the art and therefore UMM 2.0 will be based on UML 2.0. Current research is also conducted in the field of business document modeling. The current version of the UMM Add-In supports the Core Component Technical Specification 2.01. The next version of CCTS (3.0) is currently under development. It is planned that future version of the UMM Add-In also support the latest CCTS standard.

[23]

References

[1] BEA, IBM, Microsoft, SAP AG and Siebel Systems. *Business Process Execution Language for Web Services*, May 2003. Version 1.1.

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

[3] B. Hofreiter and C. Huemer. Transforming UMM Business Collaboration Models to BPEL. In *Proceedings of OTM Workshops 2004*, volume 3292, pages 507–519. Springer LNCS, 2004.

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

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

[6] H. Kim. Conceptual Modeling and Specification Generation for B2B Business Processes based on ebXML. In *SIGMOD Rec., vol. 31, no. 1, pp. 37-42*, 2002.

[7] B. Korherr and B. List. Extending the UML 2 Activity Diagram with Business Process Goals and Performance Measures and the Mapping to BPEL. In

Proceedings of the ER-Conference on Conceptual Modeling (Workshop BP-UML'06). Springer, Nov. 2006.

[8] 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.

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

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

[11] OASIS, UN/CEFACT. *ebXML - Technical Architecture Specification*, February 2001. Version 1.4.

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

[13] OMG - Object Management Group. *Business Process Modeling Notation Specification 1.0*, 2006.

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

[15] RosettaNet. *RosettaNet Implementation Framework: Core Specification*, December 2002. V02.00.01.

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

[17] S. Tyndale-Biscoe, O. Sims, B. Wood, and C. Sluman. Business Modelling for Component Systems with UML. In *Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC'02*. IEEE Press 2002, 2002.

[18] UN/CEFACT. *Electronic Data Interchange*, September 2006. D.06A.

[19] UN/CEFACT International Trade and Business Process Group (TBG14). *UN/CEFACT Common Business Process Catalog*, Nov. 2003. Version 0.95.

[20] UN/CEFACT Techniques and Methodologies Group. *UN/CEFACT's Modeling Methodology (UMM), UMM Meta Model - Foundation Module*, Oct. 2006. Technical Specification, http://www.unece.org/cefact/umm/UMM_Foundation_Module.pdf.

[21] UN/CEFACT TMG. *Core Components Technical Specification - Part 8 of the ebXML Framework*, November 2003. v2.01.

[22] UN/CEFACT TMG. *UN/CEFACT - ebXML Business Process Specification Schema*, 2003.

[23] UN/CEFACT TMG. *BCSS - UML Profile for Core Components based on CCTS 2.01*, October 2006. Candidate for version 1.0.