A 3-level e-Business Registry Meta Model

Christian Huemer, Philipp Liegl, Rainer Schuster and Marco Zapletal

Institute of Software Technology, Vienna University of Technology, Austria
{huemer, liegl}@big.tuwien.ac.at.at
{schuster, marco}@ec.tuwien.ac.at

Abstract

Business partners willing to do business electronically with each other must reach an agreement (1) on the economic level, (2) on the inter-organizational process choreography, and (3) on the services implementing the choreography. In order to search for a potential business partner, one will first look for a partner who offers a required service on the economic level and who supports a complementary role in a choreography, before binding to its IT services. Inasmuch, a registry for inter-organizational systems should cover all three levels and maintain the dependencies between them. In this paper we set up on well accepted approaches on the different levels, i.e. (1) the e^{3}value ontology, (2) the UN/CEFACT modeling methodology (UMM), and (3) the business process execution language (BPEL). We specify a registry meta model on top of ebRIM registering the artifacts on the different levels and defining their inter-dependencies.

1. Introduction

Organizations exchange numerous business documents with each other in order to do business. The idea of exchanging these documents by electronic means has existed for some decades and has been implemented by the concepts of EDI. EDI concentrated on the standardization of business documents. However, successfully implementing an inter-organizational system requires that business documents are exchanged in an agreed order. With the growing interest in Web Services, we recognize approaches to formally describing the choreography of business document exchanges.

It is desired to find potential business partners electronically according to a common choreography of business document exchanges to reach a common business goal. However, in reality most business partners meet off-line and have to undergo a cumbersome co-ordination process to align their IT-systems. The current registry-based approach does not take up because it is purely IT-driven and follows a bottom-up strategy. Each company develops its own interfaces to its own IT-system and the choreography that fits its proprietary needs. The resulting interfaces described by WSDL and the local choreography specified by BPEL are registered. In order to find a potential business partner one must look up the choreography and perform a difficult match-making process to check whether the other partner's choreography is compliant to one's own. However, if the choreographies have been developed in isolation, it is rather unlikely that they will match. Even, if the choreographies match on a technical level, there is still the risk of incompatibilities on the economic level, because the business functionality is hidden in the technical details.

For these reasons we state that the current main-stream approach to establish business partnerships on-line must be reconsidered. We motivate our approach that considers (1) the economic perspective, (2) the global choreography perspective, and (3) the IT perspective in section 2. We briefly introduce each methodology used in the corresponding perspective: the e^{3}value ontology in section 3, UN/CEFACT's Modeling Methodology (UMM) in section 4 and BPEL in section 5. In section 6 we define a registry meta-model on top of the ebXML registry model ebRIM that manages the artifacts on each level and keeps their dependencies.

2. Motivating Business Scenario

Our approach to establish business partnerships on-line is strongly motivated by ebXML [15]. One may argue that ebXML's relevance is fading due to major industries accepting Web Services as industry standard on the technical level. However, in contrast to Web Services, ebXML had a pure e-Business focus from its beginning. Thus, we feel that the basic ebXML scenario in figure 1 is still relevant, independent of the technologies used to implement this scenario.

In this scenario, company *A* wants to perform a certain business scenario. In step 1 it downloads the details of this

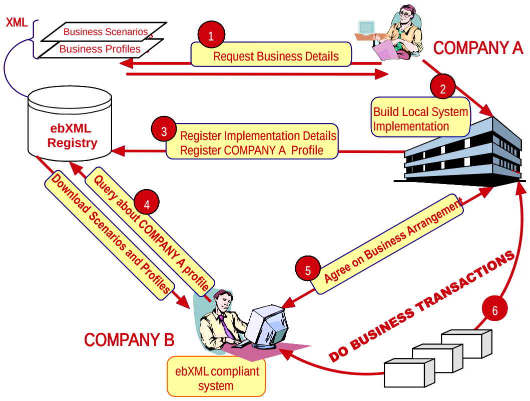

XML
Business Scenarios
Business Profiles
1
Request Business Details
COMPANY A
2
Build Local System Implementation
ebXML Registry
3
Register Implementation Details
Register COMPANY A Profile
4
Query about COMPANY A profile
Download Scenarios and Profiles
5
Agree on Business Arrangement
DO BUSINESS TRANSACTIONS
6
COMPANY B
ebXML compliant system

Figure 1. ebXML Scenario [15]

business scenario. Up to our interpretation as of today, this means it first downloads a computation-independent model describing the business functionality and business value associated with a certain scenario. As an example we assume that company *A* is a bank and looks for scenarios in the banking sector. Amongst others it finds the business scenario of a letter of credit, which is a common scenario in international trade transactions [19].

If a company recognizes a business value in a certain scenario, it will download a specification of the choreography of business document exchanges that realizes the business functionality. In our example, bank *A* sees a value in acting as an issuing bank. Thus, it will download the choreographies for interacting with an advising bank and with a buyer. This completes step 1 of the ebXML scenario.

In step 2 a company builds its local system implementation. This does not necessarily mean that it has to implement a new information system from scratch. However, it has to set up the interfaces to its local information system as required by the global choreography and specifies the local choreography for this interfaces (which may be derived from the global one). Bank *A* will specify the interfaces required for an issuing bank and the local choreographies to interact with an advising bank and with a buyer.

Having its system ready, the company will register its services in step 3. Bank *A* announces in step 3 that it is ready to act as an issuing bank in a letter of credit scenario. This means it registers its WSDL files describing the technical details of its services. Next, it registers two BPEL files describing how its services interact with an advising bank and with a buyer. Technically a single BPEL file would work, but in order to separate concerns we prefer two BPEL files. So far, bank *A* has registered its local services with the advising bank and with the buyer. To advertise the fact that these two local services are compliant to what is expected from an issuing bank in the commonly defined global choreographies, it establishes a link from each of the two local services to the corresponding global choreography within the registry.

These links facilitate the search in the registry. In step 4 company *B* is looking for a potential business partner. We assume that company *B* is a bank as well and that it is able to perform the role of an advising bank in the letter of credit scenario. Thus, it queries the registry for business partners who support the same global choreography (i.e. the one of letter of credit) and who are able to act as complementary role (i.e. as issuing bank). Accordingly, bank *B* will find bank *A* as potential business partner. Since bank *A* has established a link in the registry between the global choreography and its local service, bank *B* is able to navigate this link in order to get all the technical details for binding its services to the ones of *A* in the following steps.

This motivating scenario is based on the existence of well accepted business scenarios and global choreographies. Such scenarios and global choreographies may be developed by standardization organizations or industry consortia. It is also conceivable that they are being developed by single organizations like market leaders. In fact, it does not matter who is going to develop them. It is up-to the market to decide which scenarios become well accepted and which ones won't. It is also important to note that there will be most likely multiple global choreographies that may realize the same business scenario such as letter of credit. Some will get accepted, others won't.

Evidently, our motivating scenario refers to three different levels. The first one is the business scenario describing the economic value. In our approach we use e^3-value to describe this level. The second level describes the global choreography, i.e. the collaborative process from a neutral perspective. On this level we use UN/CEFACT's modeling methodology. Finally, the third level specifies the local services by means of BPEL and WSDL. In the following section we briefly introduce the approaches on each level.

3. e^{3}value by example

The e^{3}value methodology has been developed to model a value web consisting of actors who create, exchange, and consume things of economic value such as money, physical goods, services, or capabilities [8]. It is an ontology-based methodology for modeling and designing business models for business networks incorporating concepts from requirements engineering and conceptual modeling. e^{3}value is based on the principle of economic reciprocity meaning a "give-and-take"-approach between actors exchanging objects with an economic value - e.g. if a seller delivers goods to a buyer, he gets money in return for the goods (figure 2). The graphical notation of e^{3}value comprises a small set of concepts and relations that have been introduced by Gordijn et al. [9] [7]. In this paper we only introduce the most important concepts that are relevant for understanding our overall registry approach.

Actors are represented as rectangles in figure 2 (A). They are perceived by its environment as independent economic entities engaged in a value exchange. By exchanging *value objects* (B) they either aim for profitability (in case of an enterprise) or maximizing their economic utility (in case of an end-consumer). *Value objects* do not necessarily need to be a physical good. Sometimes they represent a service, right or even a customer guarantee. A *value object* is always modeled in combination with a *value exchange* (C) and is represented as text field. A *value exchange* is graphically modeled as a connection between *actors*.

Value objects are exchanges between *actors* using *value ports* (D). The concept of a *value port* is to signalize whether the *actor* offers or requests a *value object*. Furthermore it enables to abstract away from the internal business processes, and to focus on how external actors and other components of the e^{3}value model can be ”plugged-in”. *Value ports* are shown as small arrows pointing in the direction of the *value exchange*. A *value interface* (E) groups individual *value ports*. Each actor may have multiple *value interfaces* containing *value ports* for offering and requesting *value objects*. *Value interfaces* bundle the *value objects* an *actor* is willing to exchange in return for other *value objects*. The exchange of *value objects* via a *value interface* is atomic in order to denote reciprocity - i.e., either all exchanges occur as specified by the *value interface* or none at all.

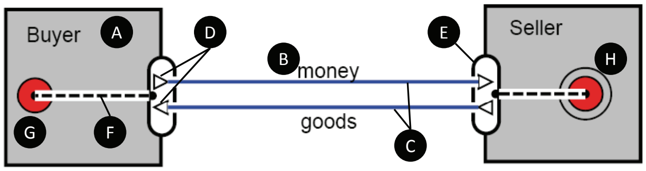


Figure 2. e3-Value notation

All concepts of e^{3}value discussed so far describe the inter-actor dependencies. In order to describe the intra-actor dependencies, scenarios are used to relate an actor’s value interface. Such scenario techniques are described by so-called use case maps (UCMs) [3] and are used within the e^{3}value methodology in a simplified way. A *scenario path* (F) indicates via which *value interfaces* objects are exchanged. In order to keep track of a *scenario path*, the *scenario path* starts with a *start stimulus* (G) and ends with a *stop stimulus* (H). With these concepts a *scenario path* can pass through different *actors* being connected by a dotted line within an *actor*. *AND forks* as well as *OR forks* (and their corresponding *joins*) can be used to model two or more sub-paths.

It is important to stress that e^{3}value does not specify any order in time. This means that there is no order between the *values exchanges* within a *value interface*. Nor is there any order between the *value exchanges* of *value interfaces* connected by *scenario paths*. This is a very significant difference between e^{3}value representing a business (value) modeling ontology and business process modeling approaches.

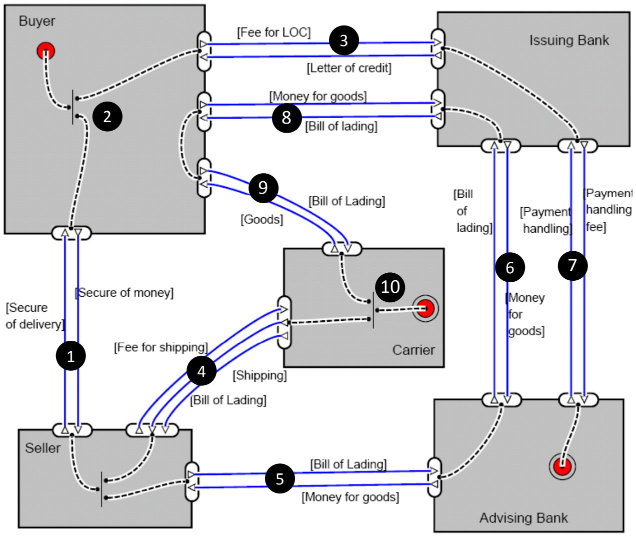


Figure 3. e3-Value example

As motivated in section 2 we demonstrate the concepts of e^{3}value by the means the *letter of credit* scenario depicted in figure 3. A detailed description of the letter of credit scenario using e^{3}value is found in [12] and [6]. The main purpose of the letter of credit is to secure the interests of both, the *seller* and the *buyer*. The seller’s economic value is to ensure, that the buyer is able to pay the goods he is delivering. Whereas the buyer’s economic value is ensuring to get the goods he is paying for. The value exchange in (1) depicts this aspect. This insurance is given by the AND-fork in (2) indicating that if the buyer wants a good, he must request a letter of credit via the value exchange (3) *AND* doing the value exchange (1) (a good for a fee). The letter of credit is a service by the *issuing bank* which ensures that, if the seller ships a good, then he gets paid.

Once the seller has the secure of money value, he ships the goods via a *carrier* in (4). In order to get the *bill of lading* (which has a significant economic value for the seller) he pays a fee for the shipping. The bill of lading is a negotiable document for the seller and is similar to paper money. Via the value exchange in (5) the seller exchanges the bill of lading for money he deserves for the goods. Furthermore the bill of lading is transferred via the issuing bank to the buyer (value exchanges (6) and (8)). The carrier delivers the ordered good to the buyer in return for the bill of lading (9). The AND-join in (10) manages that the bill of lading should also be obtained by the carrier once the good is delivered. For the services granted the issuing bank pays a payment handling fee to the advising bank as shown in the value exchange (7).

4. UN/CEFACT's Modeling Methodology

UN/CEFACT's Modeling Methodology (UMM) is defined as a UML profile consisting of stereotypes, tagged values and OCL constraints. Thereby it customizes the UML meta model to the specific needs of inter-organizational business process modeling. In contrast to the e^3value approach, UMM focuses on the process perspective and on the exchange of documents in an inter-organizational business process.

The main goal of UMM is to define a global choreography for an inter-organizational business process. A global choreography describes a process from a neutral perspective whereas a local choreography describes the process from the perspective of a participant. The global UMM choreography defines the exchange order of business documents. First the modeler specifies the specific requirements for the process in the *business requirements view*. The exact choreography of the message exchange is defined within the *business choreography view*. Finally the modeler defines the conceptual business documents exchanged in the process in the *business information view*.

A UMM model is structured according to these top-level views - figure 4 gives an overview of the package structure of UMM. In this paper only the *business choreography view* is examined in detail. The *business choreography view* is composed of one or more *business collaboration views* and one or more *business transaction views*. A business collaboration view captures the modeling elements of a business collaboration protocol. Likewise a business transaction view includes the modeling elements of a business transaction.

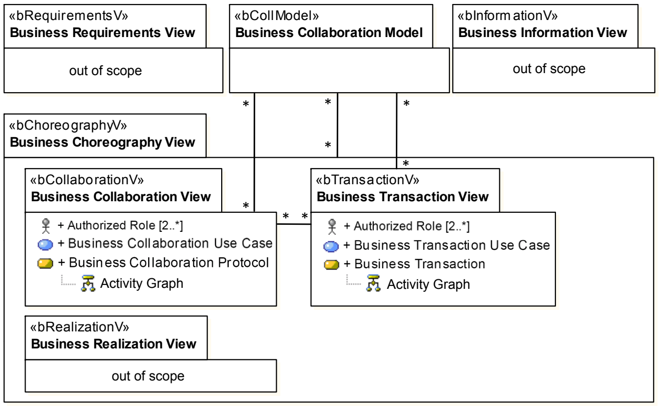

«bRequirementsV»
Business Requirements View
out of scope
«bCollModel»
Business Collaboration Model
«bInformationV»
Business Information View
out of scope
«bChoreographyV»
Business Choreography View
«bCollaborationV»
Business Collaboration View
+ Authorized Role [2..*]
+ Business Collaboration Use Case
+ Business Collaboration Protocol
Activity Graph
«bTransactionV»
Business Transaction View
+ Authorized Role [2..*]
+ Business Transaction Use Case
+ Business Transaction
Activity Graph
«bRealizationV»
Business Realization View
out of scope
*
*
*
*
*
*
*
*

Figure 4. UMM package structure

In the following we concentrate on the *business choreography view* since we want to define a registry model for business choreographies.

The exchange of a business document and an optional response between two partners is modeled using the concept of a so called *business transaction*. A dedicated *business transaction view* exists for every *business transaction*. As outlined on the right hand side of figure 4 a *business transaction view* consists of two *authorized roles*, a *business transaction use case* and a *business transaction*. The *authorized roles* are assigned to the *business transaction use case* denoting the parties involved in the business document exchange. A *business transaction use case* is further refined using the concept of a *business transaction* where the exact message exchange is described using activity diagrams.

In order to model a more complex scenario where multiple message exchanges are chained together we use the concept of a *business collaboration* shown on the left hand side of figure 4. Each *business collaboration* is stored in its own *business collaboration view*. A *business collaboration view* consists of two *authorized roles* involved in the collaboration, a *business collaboration use case* and a *business collaboration protocol*. Similar to the *business transaction* a *business collaboration protocol* further refines a *business collaboration use case*. A *business collaboration protocol* specifies a choreography amongst *business transactions*.

Figure 5 shows the concept of a UMM *business collaboration* using the letter of credit scenario. As shown in (A) the overall *business collaboration* process letter of credit between the issuing bank and the advising bank is split up into three *business transactions*: announce opening of letter of credit, request bill of lading and announce payment authorization. The *authorized roles* are connected to the *business collaboration use case* using *participates* associations.

In (B) the use case for the *business transaction* announce payment authorization is shown involving the two *authorized roles* announcer and announcee. As mentioned before, the use case is further refined using a *business transaction* as shown in (D). Using the concept of activity diagram swimlanes the two *authorized roles* announcer and announcee are shown. The announcer sends a payment authorization request envelope to the announcee who in turn either replies with a payment rejected envelope or a payment accepted envelope. The *business transaction* as shown in (D) always follows the same pattern employing two swimlanes, two actions, a requesting document, optional responding documents and so called *business entity states*. The latter ones mark the result of a *business transaction*.

In (C) the *business transaction* announce payment

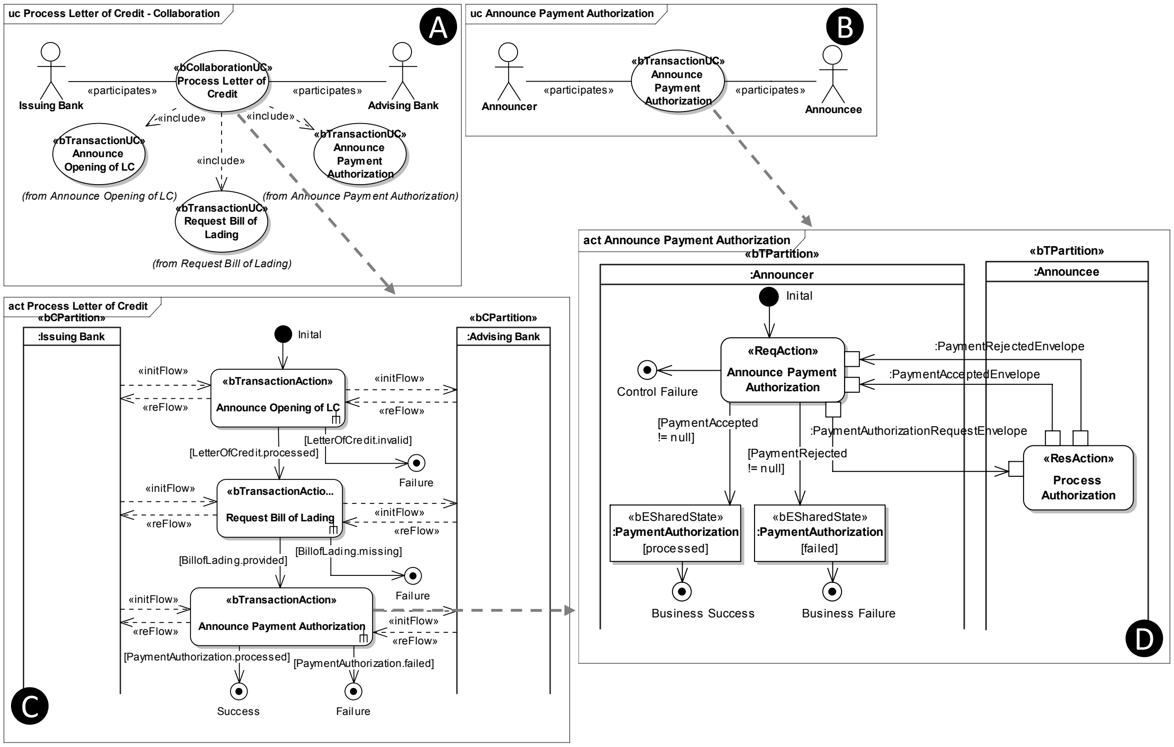

uc Process Letter of Credit - Collaboration
A
Issuing Bank
«participates»
«bCollaborationUC» Process Letter of Credit
«participates»
Advising Bank
«include»
«include»
«include»
«bTransactionUC» Announce Opening of LC
(from Announce Opening of LC)
«bTransactionUC» Request Bill of Lading
(from Request Bill of Lading)
«bTransactionUC» Announce Payment Authorization
(from Announce Payment Authorization)
uc Announce Payment Authorization
B
Announcer
«participates»
«bTransactionUC» Announce Payment Authorization
«participates»
Announcee
act Process Letter of Credit
«bCPartition» :Issuing Bank
«bCPartition» :Advising Bank
Inital
«initFlow»
«reFlow»
«bTransactionAction» Announce Opening of LC
[LetterOfCredit.invalid]
[LetterOfCredit.processed]
Failure
«bTransactionActio... Request Bill of Lading
[BillofLading.missing]
[BillofLading.provided]
«bTransactionAction» Announce Payment Authorization
[PaymentAuthorization.processed]
[PaymentAuthorization.failed]
Success
Failure
C
act Announce Payment Authorization
«bTPartition» :Announcer
«bTPartition» :Announcee
Inital
«ReqAction» Announce Payment Authorization
Control Failure
:PaymentRejectedEnvelope
:PaymentAcceptedEnvelope
:PaymentAuthorizationRequestEnvelope
[PaymentAccepted != null]
[PaymentRejected != null]
«ResAction» Process Authorization
«bESharedState» :PaymentAuthorization [processed]
«bESharedState» :PaymentAuthorization [failed]
Business Success
Business Failure
D

Figure 5. UMM by example - Letter of Credit

authorization is put into the global choreography of the letter of credit process. As shown in (C) the collaboration starts with announce opening of letter of credit followed by request bill of lading. If these two transactions are successful the third and last transaction is reached - announce payment authorization. The different parties involved in the *business collaboration* are modeled using swimlanes.

In particular a *business transaction* (D) is a strong candidate for reuse. Since it always follows the same pattern, a modeler may retrieve information of an existing transaction from the registry and reuse it in multiple application scenarios. Furthermore, the business collaboration process letter of credit as shown in (A) and (C) may be part of a more complex collaboration. Since business collaborations can be nested, the letter of credit collaboration may be integrated in a larger application scenario.

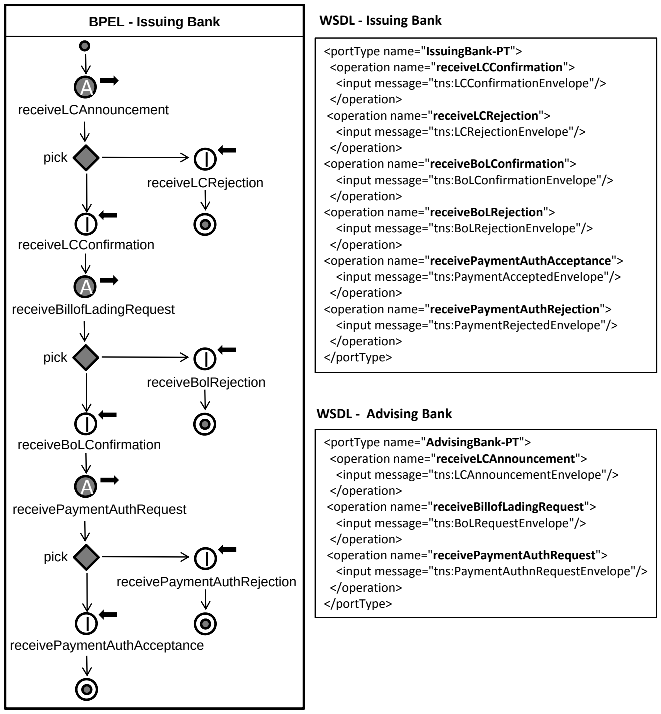
In regard to the registry meta model we can identify the following prerequisites as shown in 4: we will store a *business collaboration model* together with the two main views for a collaboration: the *business collaboration view* and the *business transaction view* with the respective artifacts of the packages. The embracing *business choreography view* will not be stored since no additional value is provided by it. In order to allow for the necessary dependencies between the three packages *business collaboration model*, *business transaction view* and *business collaboration view* we use the concept of references. The references between the different packages are indicated by associations in figure 4.

5. A local choreography using BPEL

The previous section showed how UMM is used to align the order of the business documents exchanged in a business collaboration. It provides the foundation for deriving software artifacts such as BPEL code, realizing the implementation of a business partner's service interface. Based on the global choreography as outlined in (C) in figure 5 a local choreography is derived for each participating business partner. The local choreographies serve as blueprints to check local B2B implementations for compliance with the agreed collaborative process. The derivation of local BPEL choreographies from standardized global UMM choreographies may be applied automatically according to general and standardized mapping rules. The interested reader is referred to our work in [10], which describes the derivation in detail.

As shown in (C) in figure 5 the letter of credit collaboration is a sequence of three *business transactions*. In each *business transaction* the *issuing bank* sends a request to the *advising bank* which replies with a response.

In figure 6 the local choreography for the *issuing bank* is shown on the left hand side using a conceptual BPEL notation. Black dots denoted with *A* indicate, that the *issuing bank* is calling an operation of the port type of the *advising bank*. White dots denoted with *I* refer to operations of the port type of the issuing bank. They indicate, that the *issuing bank* is waiting for a message of the *advising bank*.



```
<portType name="IssuingBank-PT">
  <operation name="receiveLCConfirmation">
    <input message="tns:LCConfirmationEnvelope"/>
  </operation>
 <operation name="receiveLCRejection">
    <input message="tns:LCRejectionEnvelope"/>
  </operation>
<operation name="receiveBoLConfirmation">
    <input message="tns:BoLConfirmationEnvelope"/>
  </operation>
<operation name="receiveBoLRejection">
    <input message="tns:BoLRejectionEnvelope"/>
  </operation>
<operation name="receivePaymentAuthAcceptance">
    <input message="tns:PaymentAcceptedEnvelope"/>
  </operation>
<operation name="receivePaymentAuthRejection">
    <input message="tns:PaymentRejectedEnvelope"/>
  </operation>
</portType>
```

```
<portType name="AdvisingBank-PT">
  <operation name="receiveLCAnnouncement">
    <input message="tns:LCAnnouncementEnvelope"/>
  </operation>
 <operation name="receiveBillofLadingRequest">
    <input message="tns:BoLRequestEnvelope"/>
  </operation>
 <operation name="receivePaymentAuthRequest">
    <input message="tns:PaymentAuthnRequestEnvelope"/>
  </operation>
</portType>
```

Figure 6. Local choreography - Issuing Bank

On the right hand side the `portTypes` of the *issuing* and of the *advising bank* together with the offered `operations` per business partner are shown using a simplified WSDL notation.

As shown on the left hand side of figure 6 the process starts with the *issuing bank* calling the `receive letter of credit announcement` operation of the *advising bank*. Since the BPEL process in figure 6 is asynchronous the *issuing bank* then waits for the *advising bank* to reply with either a `letter of credit rejection` or a `letter of credit confirmation`. In the former case the process terminates. In the latter case the *issuing bank* calls the `receive bill of lading request` operation of the *advising bank* and again waits for a confirmation or a rejection. If the `bill of lading confirmation` has been received successfully a `payment authorization request` is sent to the *advising bank*. If the *advising bank* replies with a `payment authorization acceptance` the process terminates successfully.

The same local choreography matching the one of the issuing bank is derived for the advising bank. Due to space limitations the advising bank's local choreography is not shown.

6. The e-Business registry meta model

Having introduced the methodologies of the three layers, we now discuss how the artifacts created by these methodologies are managed in a registry. For this purpose we propose a registry meta model based on the ebXML registry information model [17] that supports the specifics of the artifacts on the three layers. Our e-Business registry meta model has the purpose of defining which artifacts are maintained in the registry. An ebXML registry stores these artifacts as *extrinsic objects*, which are XML and RDF files in our case, but may also be binaries in a general case. The content of the *extrinsic object* is encapsulated - this means a query to the registry does not access the content of an *extrinsic object*. Thus, an *extrinsic object* must be associated with appropriate meta data to allow an effective search. Our e-Business registry meta model defines appropriate meta data for each of the artifacts on the three levels. Furthermore, the different artifacts and their meta data have dependencies on each other. The e-Business registry meta model defines the required links between the *extrinsic objects* of the different artifacts and also between their meta data if appropriate.

The resulting meta model is depicted in figure 7. *Extrinsic objects* representing the various artifacts are shown with a thick border. The classes referring to the value layer of e^{3}value are presented with gray background. The ones managing the global choreography of UMM are depicted with white background. Finally the ones relating to the Web Services deployment layer are presented with black background. Each class of our e-Business registry model is based on an existing meta class of ebRIM. The corresponding ebRIM meta class is denoted in the upper right corner of each class.

Registering e^{3}value models: In section 3 we presented the graphical representation of e^{3}value models. In addition to this graphical notation, e^{3}value models may be expressed by means of RDF [1]. This RDF representation is used to store the e^{3}value model as an extrinsic object in the e-Business registry. In a next step we have to decide on the meta data to accompany the e^{3}value. Evidently, the definition capturing the business justification of an e^{3}value model is a kind of meta data. Additionally, the participants of a business network as well as the exchanged objects of value may be of interest when searching for potential business partners. Correspondingly, we extract this information from an e^{3}value model and use it for annotating the *e3-ValueModel*. The definition as well as each actor of the *e3-ValueModel* are represented by their own *slots* and con-

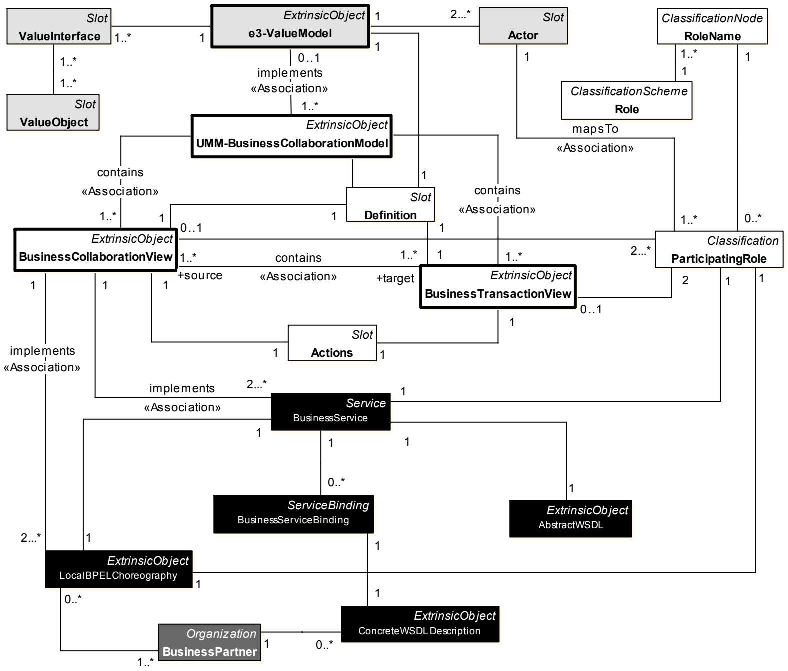


Figure 7. e-Business registry meta model

nected to the *e3-ValueModel*. In a similar way, we map value interfaces and the exchanged value objects to slots in our e-Business registry meta model. In ebRIM, the concept of *slots* is used to add arbitrary meta data to registry objects

Registering UMM models: An UMM model is based on the UML syntax. The graphical UML syntax may also be represented in XMI. In [11], we discuss how to represent a UMM 1.0 model by means of XMI. The XMI representation of a UMM model is used to store it as an extrinsic object - *UMM-BusinessCollaborationModel* - in the registry. In order to foster re-use of UMM artifacts, our approach does not only support registration of a whole UMM model, but also of parts thereof. As we outlined in section 4, UMM business transactions and business collaborations may be re-used across different UMM models. Hence, we consider the storage of business collaboration views and business transaction views as self-contained units. Each of them is represented by its own extrinsic object. A business registry that implements the e-Business registry meta model has to extract the XMI of the respective parts of a UMM model and has to store them as separate entities. For keeping the relationships between the UMM model and its business collaborations and business transactions, associations between a *UMM-BusinessCollaborationModel* and its *BusinessCollaborationViews* and *BusinessTransactionViews* must be created. A slot named *Actions* is assigned to each *BusinessCollaborationView* and *BusinessTransactionView* for textually describing the flow of actions. Furthermore, we introduce a slot called *Definition* to capture the purpose and the business justification of a UMM model.

In addition to textual descriptions of process flows, the roles in a business process are candidates for being stored as meta data. Accordingly, we link each *BusinessCollaborationView* and each *BusinessTransactionView* with their corresponding authorized roles. Since business transactions describe binary relationships, exactly two *ParticipatingRoles* are connected to a *BusinessTransactionView*. Since UMM business collaborations may also capture multi-party processes, two or more *ParticipatingRoles* may be connected to a *BusinessCollaborationView*. The roles that participate in a standardized UN/CEFACT business collaboration should be named according to a given scheme. UN/CEFACT currently identifies a common set of role names for business collaborations. In order to represent such a taxonomy for roles in our e-Business registry meta model, we employ the concepts of *classification scheme* and *classification nodes*. The classification *ParticipatingRole* expresses that a given role takes part in a certain business collaboration or business transaction.

Linking e^{3}value and UMM models: In order to offer business partners the possibility to query for a business collaboration model that fulfills a certain business model, those types of models must be linked in a business registry. Accordingly, the *UMM-BusinessCollaborationModel* and the *e3-ValueModel* are connected with an association. ebRIM requires that an association between two registry

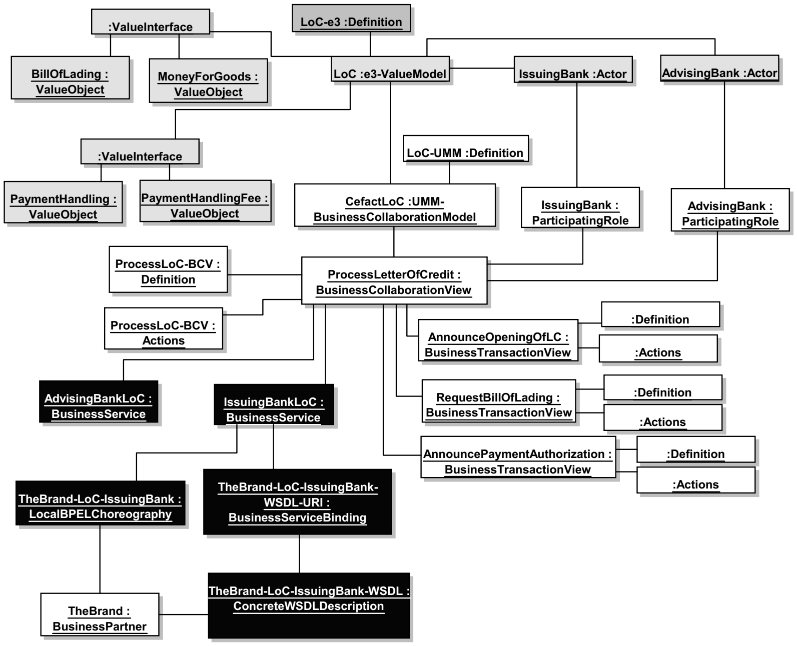

:ValueInterface
LoC-e3 :Definition
BillOfLading : ValueObject
MoneyForGoods : ValueObject
LoC :e3-ValueModel
IssuingBank :Actor
AdvisingBank :Actor
:ValueInterface
LoC-UMM :Definition
PaymentHandling : ValueObject
PaymentHandlingFee : ValueObject
CefactLoC :UMM-BusinessCollaborationModel
IssuingBank : ParticipatingRole
AdvisingBank : ParticipatingRole
ProcessLoC-BCV : Definition
ProcessLetterOfCredit : BusinessCollaborationView
ProcessLoC-BCV : Actions
:Definition
AnnounceOpeningOfLC : BusinessTransactionView
:Actions
AdvisingBankLoC : BusinessService
IssuingBankLoC : BusinessService
:Definition
RequestBillOfLading : BusinessTransactionView
:Actions
AnnouncePaymentAuthorization : BusinessTransactionView
:Definition
:Actions
TheBrand-LoC-IssuingBank : LocalBPELChoreography
TheBrand-LoC-IssuingBank-WSDL-URI : BusinessServiceBinding
TheBrand-LoC-IssuingBank-WSDL : ConcreteWSDLDescription
TheBrand : BusinessPartner

Figure 8. e-Business Registry Example - Letter of Credit

objects must have a certain type that identifies the type of association. For the purpose of our e-Business registry model two generic association types are sufficient: *implements* and *contains*. Since business models are implemented by business processes, we use the *implements* association between the entities *e3-ValueModel* and *UMM-BusinessCollaborationModel*.

Moreover, as shown in figure 7 we introduce the association type *mapsTo* for connecting roles on the choreography level with actors on the value level. The *mapsTo* association indicates which role in the business collaboration is fulfilled by which actor in the value network - and vice versa.

Registering Web Services artifacts: We map deployment artifacts - local BPEL choreographies and WSDL artifacts - to the e-Business registry model and link them to their graphical UMM representations. The implementation of a global UMM collaboration requires commitments of the participating business partners in terms of service offerings. In other words, each participant has to implement his part of the process and has to expose this implementation as a service to his partners. Correspondingly, a UMM business collaboration is implemented by two or more *BusinessServices* - one per each participant. ebRIM provides the generic type *Service* for representing business services in our e-Business registry meta model. One should note, that we use the term service in this context as a business economic activity (mostly intangible in nature), offered by one party to another in order to achieve a certain benefit [22]. In order to foster the implementation of standardized business services, abstract WSDL definitions may be provided by UN/CEFACT. An *AbstractWSDL* captures the implementation specifics of a service that has to be offered by a participant to support a certain business collaboration. Consequently, we establish a relationship between *BusinessService* and *AbstractWSDL* in our e-Business registry meta model as shown in figure 7.

In addition to abstract WSDL descriptions, UN/CEFACT may supply standardized local BPEL choreographies describing the flow of message exchanges with respect to each participant. We already know that a business collaboration in UMM may be either binary or multi-party. Thus, two or more *LocalBPELChoreographies* may implement the message exchange as described in a global *BusinessCollaborationView*.

Registering companies and their services: As motivated in section 2 companies may present themselves within a business registry. In our e-Business registry meta model we use the ebRIM type *Organization* to represent concrete companies. A set of additional ebRIM types like person name, postal address, email, etc. may be used to detail a company's profile. Since the usage of these aspects is considered as straight-forward we do not include them in our proposed e-Business registry meta model. Once a company is registered it specifies which business collaborations it supports in which role. In our e-Business registry meta model, this is simply specified by establishing a relationship between a *BusinessPartner* and a *LocalBPELChoreography*. Furthermore, companies may register concrete

WSDL descriptions for the *BusinessServices* they support. A partner's *ConcreteWSDLDescription* is linked to a *BusinessService* via the entity *BusinessServiceBinding*. The ebRIM concept of a service binding contains the URI, where an implementation of a service may be located. In the e-Business registry meta model, the *BusinessServiceBinding* contains the URI of the *BusinessService* that is offered by a certain company. The *ConcreteWSDLDescription* holds the required information (e.g., protocol bindings) to access the respective business service and to bind it at run-time.

e-Business registry example: Having introduced all concepts of our e-Business registry example, we illustrated these concepts in figure 8 by means of our `letter of credit` example. An RDF equivalent of the e^{3}value model in figure 3 is stored as an extrinsic object in the registry. The meta data associated to this extrinsic object are the slots for its definition and for the actors `issuing bank` and `advising bank`. Furthermore is is associated with two value interfaces. The first one covering bill of lading in exchange for money for goods, and the second one being composed of payment handling in return for a payment handling fee. Note, the overall example would consist of more actors and value interfaces, but we show only those registry elements that are meaningful for a collaboration between the issuing bank and the advising bank to keep the model simple.

The e^{3}value model is associated with the XMI representation of a UMM business collaboration model for `letter of credit`. The meta data of this business collaboration model includes slots for its definition and for the participating roles `issuing bank` and `advising bank`. Mappings between the e^{3}value actors `issuing/advising bank` and the corresponding UMM participating roles are established. Furthermore, the business collaboration view `process letter of credit` between the `issuing bank` and the `advising bank` as well as the business transaction views being part of this collaboration are extracted and stored as separate extrinsic objects. The corresponding relationships are maintained in the registry. Furthermore, slots for the definitions and actions containing the meta data of these views are created.

The business collaboration view `process letter of credit` covers a global choreography described from a neutral perspective. The deployment artifacts on each business partner's side are local choreographies described from the corresponding partner's perspective. Accordingly, there is a business service for the `letter of credit` scenario for the `issuing bank` and another one for the `advising bank`. We further detail only the one of the issuing bank.

A company ('The brand') registers itself as a business partner. It furthermore registers its local choreography for acting as an `issuing bank`. This local choreography is composed of operation calls specified in a (or even more) WSDL file(s). The company also registers the WSDL file. In order to advertise the fact that its interfaces meet the requirements of the overall process it links the BPEL file and the WSDL file via a business service binding to the business service of the `issuing bank`. Thereby, its local implementation is also bound transitively to the global choreography of the business collaboration view and, finally, to the e^{3}value model.

7. Related Work

Research in the field of registry models and registry classifications has brought a set of challenging approaches. In [13] an ontology driven registry classification model is presented. Using OWL [18] Liu et al. describe an ontology for an ebXML registry. Thereby they abstract from the original ebRIM specification by describing the registry classification using OWL in order to retrieve a registry classification model from it.

Another approach using OWL to extend ebRIM is presented in [21] and [4]. Roh et al. are using OWL to build a foundation for intelligent information processing in ebXML registries. Thereby they propose a new ebXML registry information model called *semantic information model* (SIM). Dogac et al. are presenting a mapping mechanism between the various constructs of OWL and the ebXML classification hierarchy.

Another extension model for the ebXML registry is presented in [14], focusing on the enhancement of the ebXML registry in order to store non-ebusiness related artifacts. The approach presented specifically focuses on federated registries and points out the flaws of the current ebRIM approach in regard to a federated registry environment. In order to cope with the challenge of a federated environment and semantics, two new types of registry objects are proposed - *SimpleObject* and *ComplexObject*. The proposed solution uses the Meta Object Facility (MOF) to redefine and extend the ebRIM standard. The meaning of the original ebRIM elements is preserved hence ensuring backwards compatibility of the presented solution to the original standard.

None of these ebRIM extensions however is specifically dedicated to the storage and retrieval of artifacts spanning the economic level, the business process level and the deployment level.

In the field of UDDI-based registries [16] (Universal Description, Discovery and Integration) [2] proposes an extension to the current UDDI standard called UDDIe. Using the concept of so called *blue pages* the UDDIe allows to store user defined properties associated with a specific service. This greatly enhances the process of service discovery and retrieval since additional meta-information can be stored in the registry.

8. Conclusion and future work

In this paper we have shown a three level approach for an e-Business registry meta model combining the economic, the process choreography, and the service implementation layers. Based on ebRIM a meta model for the storage and retrieval of artifacts in an ebXML registry has been developed. Using the concept of ebRIM *slots* we have built a meta model which facilitates the search for modeling and technical artifacts. The modeler can either search for e3value artifacts on the business level or for UMM business transactions and collaborations artifacts on the process model level. On the technical level the user can retrieve WSDL and BPEL information for existing business process models.

Currently, our approach is bound to a certain methodology on each layer. Future work has to be directed towards flexibility in regard to the supported artifacts which can be stored in the registry. By altering the *extrinsic objects* the designed meta model should support different methodologies. For example, one may think of supporting REA models [5] at the business level or WS-CDL [20] artifacts at the choreography level.

Another future work item is the enrichment of the designed meta model with semantic technologies. Although the search for definitions, actors, value interfaces etc. provides an effective search means for the modeler, a future extension with semantic technologies is planned. A soft spot of our meta data support is the *definition slot* which is currently plain text based. We plan to extend the concept of a *definition slot* with additional semantic extensions in future work. Thereby a method must be found allowing to describe the definition slot by an appropriate ontology, such as OWL [18].

References

[1] H. Akkermans, Z. Baida, J. Gordijn, N. Peña, A. Altuna, and I. Laresgoiti. Value webs: Using ontologies to bundle real-world services. *IEEE Intelligent Systems*, 19(4):57–66, 2004.

[2] A. S. Ali, O. F. Rana, R. Al-Ali, and D. W. Walker. UDDIe: an extended registry for Web services. In *Symposium on Applications and the Internet*, 2003.

[3] R. J. A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Trans. Software Eng.*, 24(12):1131–1155, 1998.

[4] A. Dogac, Y. Kabak, and G. B. Laleci. Enriching ebXML Registries with OWL Ontologies for Efficient Service Discovery. In *14th International Workshop on Research Issues on Data Engineering*, 2004.

[5] G. L. Geerts and W. E. McCarthy. An accounting object infrastructure for knowledge-based enterprise models. *IEEE Intelligent Systems*, 14(4):89–94, 1999.

[6] J. Gordijn. A design methodology for modeling trustworthy value webs. *Int. J. Electron. Commerce*, 9(3):31–48, 2005.

[7] J. Gordijn and H. Akkermans. Designing and evaluating e-business models. *IEEE Intelligent Systems - Intelligent e-Business*, 16:11–17, 2001.

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

[9] J. Gordijn, H. Akkermans, and H. van Vliet. Business modelling is not process modelling. In *ER (Workshops)*, pages 40–51, 2000.

[10] 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.

[11] B. Hofreiter, C. Huemer, and M. Zapletal. Registering UMM Business Collaboration Models in an ebXML Registry. In *8th IEEE International Conference on E-Commerce Technology (CEC 2006) / Third IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE 2006)*, 2006.

[12] V. Kartseva, J. Gordijn, and Y.-H. Tan. Analysing preventative and detective control mechanisms in international trade using value modelling. In *ICEC*, pages 51–58, 2004.

[13] W. Liu, K. He, and W. Liu. Design and realization of ebXML registry classification model based on ontology. In *International Conference on Information Technology: Coding and Computing (ITCC)*, volume 2, pages 809–814, 2005.

[14] W. Liu, L. He, J. Liu, and K. He. A semantic interoperability extension model to the ebXML registry. In *International Conference on Information Technology: Coding and Computing (ITCC)*, 2005.

[15] OASIS. *ebXML Technical Architecture Specification*, Feb. 2001. Specification 1.0.4.

[16] OASIS. *UDDI Specification*, Oct. 2004. Technical Committee Draft, Version 3.0.2.

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

[18] W3C. *Web Ontoloty Language (OWL)*. W3C.

[19] Wikipedia. Letter of Credit, 2007. `http://en.wikipedia.org/wiki/Letter_of_credit` [accessed 10-Feburary-2007].

[20] World Wide Web Consortium (W3C). *Web Services Choreography Description Language*, Nov. 2005. Version 1.0, W3C Candidate Recommendation.

[21] R. Yohan, K. Hangkyu, K. H. Soo, K. M. Ho, and S. J. Hyun. Semantic business registry information model. In *International Conference on Convergence Information Technology*, 2007.

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