Process Models and Business Models – a Unified Framework

Maria Bergholtz, Prasad Jayaweera, Paul Johannesson, Petia Wohed

Department of Computer and Systems Sciences
Stockholm University and Royal Institute of Technology
Forum 100, SE-164 40 Kista, Sweden

e-mail: {maria, prasad, pajo, petia}@dsv.su.se

Abstract. In e-Commerce, there are two fundamental types of models, business models and process models. A business model is concerned with value exchanges among business partners, while a process model focuses on operational and procedural aspects of business communication. Thus, a business model defines the *what* in an e-Commerce system, while a process model defines the *how*. The purpose of this paper is to analyse the contents of business models and process models and to show how they can be integrated. We are using ebXML as a conceptual and notational framework for our approach. The theoretical foundations of our approach are based on the Language/Action approach and REA. We illustrate how our approach can be used to facilitate integration, process specification and process pattern interpretation.

1 Introduction

With the growing interest and activities in e-Commerce, there is an increasing need for methods and techniques that can help in the design and management of e-Commerce systems. In e-Commerce, systems design is based on two fundamental types of models, business models and process models. A business model is concerned with value exchanges among business partners, while a process model focuses on operational and procedural aspects of business communication. Thus, a business model defines the *what* in an e-Commerce system, while a process model defines the *how*. This means that the process of designing e-Commerce systems consists of two main phases. First, a business requirement capture phase focusing on value exchanges, and secondly, a phase focused on operational and procedural realisation.

In the business requirement capture phase, coarse-grained views of business activities as well as their relationships and arrangements in business collaborations are represented by means of business model constructs at an abstract level. In contrast, the specification of a process model deals with more fine-grained views of business transactions, their relationships and choreography in business collaborations. Although the two phases in e-Commerce design, and their related models, have different focuses, there is clearly a need for integrating them. A unified framework covering coarse-grained business modelling views to fine-grained process specification views provides several benefits. It can be used for supporting different user views of the system being designed, and it can form the basis of a precise understanding of modelling views and their inter-relationships. Another advantage of a unified framework is that it can be used for process integration, i.e. to provide measures for the establishment of correspondences between different structures in process models.

The purpose of this paper is to analyse the contents of business models and process models and to show how they can be integrated. We use ebXML [**10**] as a conceptual and notational framework for our approach, more specifically BPSS (Business Process Specification Schema) for process models and UN/CEFACT UMM [**22**] for business models. The theoretical foundations of our approach are based on the Language/Action approach and REA.

The rest of the paper is organised as follows. Section 2 gives an overview of related research. Section 3 introduces the UMM Business Requirement View. Section 4 describes process models according to BPSS. Section 5 contains the main contribution of the paper and shows how to integrate business and process models based on a Language/Action approach. Section 6 illustrates two applications of the introduced framework. Section 7 presents conclusions and suggests future research directions.

2 Related Research

The approach proposed in this paper is based on elements from the Language Action approach and the REA ontology [**15**]. The Language Action approach to information systems design (based on speech act theory [**1**]), focuses on communication aspects when analysing and developing a system. A speech act is defined as an action changing the universe of discourse when a speaker utters it and a recipient grasps it. It may be oral as well as written, or even expressed via some other communication form such as sign language. Searle has developed speech act theory [**20**] by introducing a taxonomy of five different kinds of speech acts: assertive, directive, commissive, expressive, and declarative, also called illocutionary points.

An *assertive* is a speech act the purpose of which is to convey information about some state of affairs of the world from one agent, the speaker, to another, the hearer. For example, the utterance "The father of speech act theory was Austin". A *commissive* is a speech act, the purpose of which is to commit the speaker to carry out some action or to bring about some state of affairs. An example is the utterance "I will complete and submit the paper to eCOMO02". A *directive* is a speech act, where the speaker requests the hearer to carry out some action or to bring about some state of affairs, e.g.. "You can complete and submit the paper to eCOMO02". A *declarative* is a speech act, where the speaker brings about some state of affairs by the mere performance of the speech act. An example is the establishment of accepted papers, e.g. "Paper no 23 is accepted". Finally, an *expressive* is a speech act, the purpose of which is to express the speaker's attitude to some state of affairs, e.g. "I like the ideas presented in this paper".

In addition to its illocutionary point, a speech act also has a propositional content. For instance, the speech acts “ I hereby pronounce you husband and wife” and “You are hereby divorced”, which are both declaratives, have different propositional contents. Furthermore, speech acts with different illocutionary points may have one and the same propositional content, which is the case with the examples for directive and commissive given above. A speech act is often viewed as consisting of two parts, its propositional content and its illocutionary force. The illocutionary force is the illocutionary point together with the manner (for example ordering, asking, begging) in which the speech act is performed and the context in which it occurs.

Some well-known and recent language action approaches are Action Workflow [**16**], Business Action Theory (BAT) [**7**], and Dynamic Essential Modelling of Organisations (DEMO) [**2**].

The second building stone of our approach, the Resource-Event-Agent (REA) framework, [**21**], has been designed for representing and reasoning about economic phenomena, more specifically about economic exchanges. REA was originally conceived as a framework for accounting systems, but it has subsequently broadened its scope and developed into an enterprise domain ontology.

The REA framework is based on three main components: Economic Agents, Economic Resources, and Economic Events, see Fig. 1. An Economic Agent is a person or organisation that is capable of controlling Economic Resources and interacting with other Economic Agents. An Economic Resource is something, e.g. goods or money, that is viewed as being valuable by Economic Agents. An Economic Event is the transfer of control of an Economic Resource from one Economic Agent to another one.

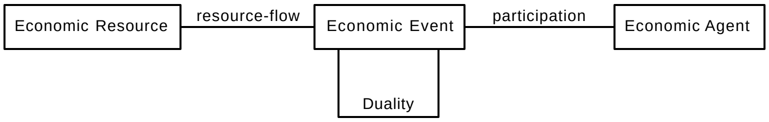

Fig. 1. Resources, Events and Agents (REA)

A central component in REA is the *Duality* existing between two Economic Events, i.e. one agent transfers some resource to another agent and receives in return another resource from that agent. This *Duality* of resource transfer is essential in commerce. It never happens that one agent simply gives away a resource to another agent without expecting another resource back as compensation.

3 UMM Business Requirements Views

The REA framework has recently been applied in the UN/CEFACT Modelling Methodology (UMM) [**22**], where it is used as a theoretical foundation of the Business Requirements View. UMM is based on the Unified Modelling Language (UML) [**18**], and it provides a procedure for modelling business processes in a technology-neutral, implementation-independent manner. In UMM, a number of different view meta-models are defined to support an incremental model development and to provide different levels of specification granularity. Among these is the Business Requirement View (BRV), see Fig. 2, capturing the business transactions with their interrelationships, which makes it the most relevant meta-model for our work.

Like REA, BRV models Economic Events, the Economic Resources transferred through the Economic Events, and the Economic Agents, here called Partner Types between whom the Economic Events are performed. Furthermore, an Economic Event fulfils an Economic Commitment. An Economic Commitment can be seen as the result of a commissive speech act and is intended to model an obligation for the performance of an Economic Event. The *duality* between Economic Events is inherited into the Economic Commitments, represented by the relationship *reciprocal*.

In order to represent collections of related commitments, the concept of Economic Contracts is used. An Economic Contract is an aggregation of two or more reciprocal commitments. An example of an Economic Contract is a purchase order with several order lines, which are the Economic Commitments, involved in the purchase order contract. The products specified in each line are the Economic Resource Types that are the subject for the Economic Commitments.

Moving one level up, the Economic Contracts are often made within the boundaries of different Agreements. An Agreement is an arrangement between two Partner Types that specifies the conditions under which they will trade, e.g., terms of shipment, terms of payment, etc. An agreement is considered to not directly imply any commitments, it rather regulates the trade conditions between partners.

Furthermore, a Business Collaboration choreographs the Business Collaboration Task performed in a contract formation when the contract formation requires a number of requesting and responding business interactions. For instance, the creation of a “purchase order request” can be specified as a Business Collaboration that choreographs both a “purchase order” and “notification of acceptance” Business Collaboration Task.

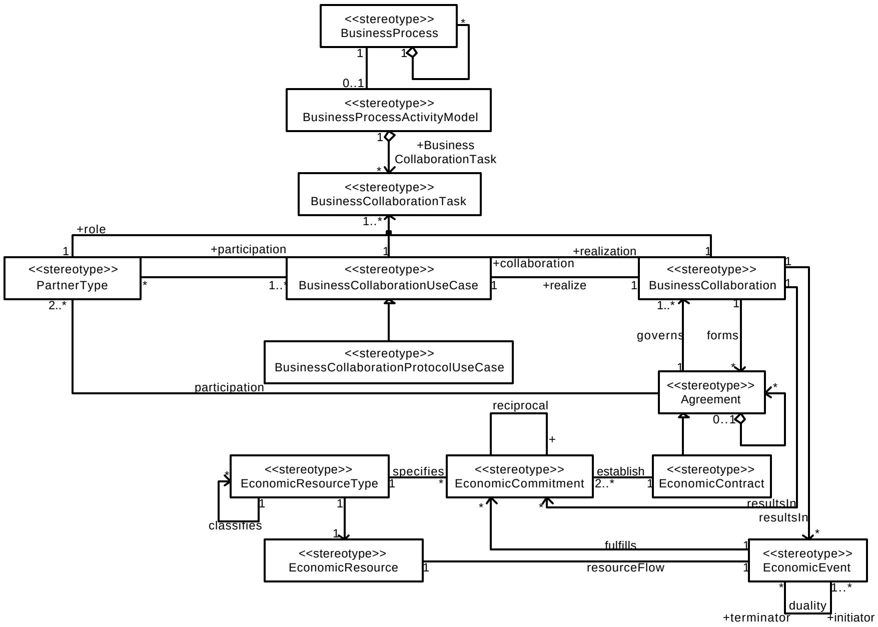

Fig. 2. UMM Business Requirements View (BRV)

4 Business Process Specification Schema

There are well-established modelling techniques for e-Commerce development. However, there is still a considerable gap to be covered between e-Commerce process modelling and specification of software components. One major framework intended to bridge this gap is the ebXML Business Process Specification Schema (BPSS), [**4**]. In BPSS, a collaboration consists of a set of business transactions choreographed to obtain a control flow of the transactions. Each transaction consists of a pair of request and response activities carried out by two business partners in different roles as well as document flows between the activities, see Fig. 3. A collaboration may be between two parties, a Binary Collaboration, or between several parties, a Multi Party Collaboration. A UML class diagram of BPSS is shown in Fig. 4.

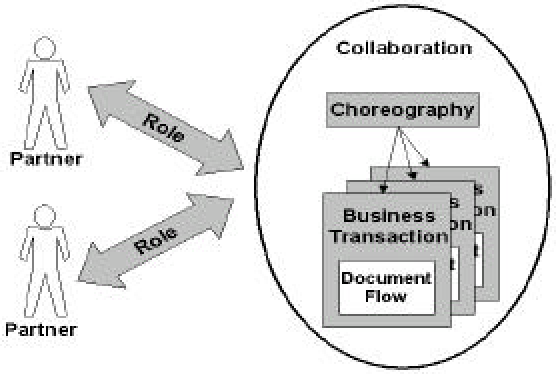

Fig. 3. Collaborations in ebXML (reprinted from the ebXML specification [**4**])

A Business Transaction is an atomic unit of work; it consists of one Requesting Business Activity, one Responding Business Activity, and one or two document flows between them. There is always a Request document flow, while there does not have to be a corresponding Response document flow. A pair of Request and Response document flows is needed in cases where some kind of agreement is to be established. Some transactions, however, have the function of notifications, and in such cases only a Request document flow is needed. There is a common superclass, Business Action, for Requesting Business Activity and Responding Business Activity, which holds common attributes specifying conditions on intelligibility checks, authorisation, time to acknowledge, and non-repudiation. An example of a Requesting Business Activity is "Request Purchase Order", an example of a Responding Business Activity is "Accept Purchase Order" – together these two Business Actions constitute a Business Transaction.

Business Transactions are the basic building blocks of Binary Collaborations. A Binary Collaboration is always between two roles, and it consists of one or more Business Activities. These Business Activities are always conducted between the two roles of the Binary Collaboration. One of these roles is assigned to be the *initiatingRole* (from) and the other to be the *respondingRole* (to). An example of a Business Collaboration is "Manage Purchase" which could involve several Business Activities for querying about products, establishing a purchase order, and establishing the individual purchase order lines.

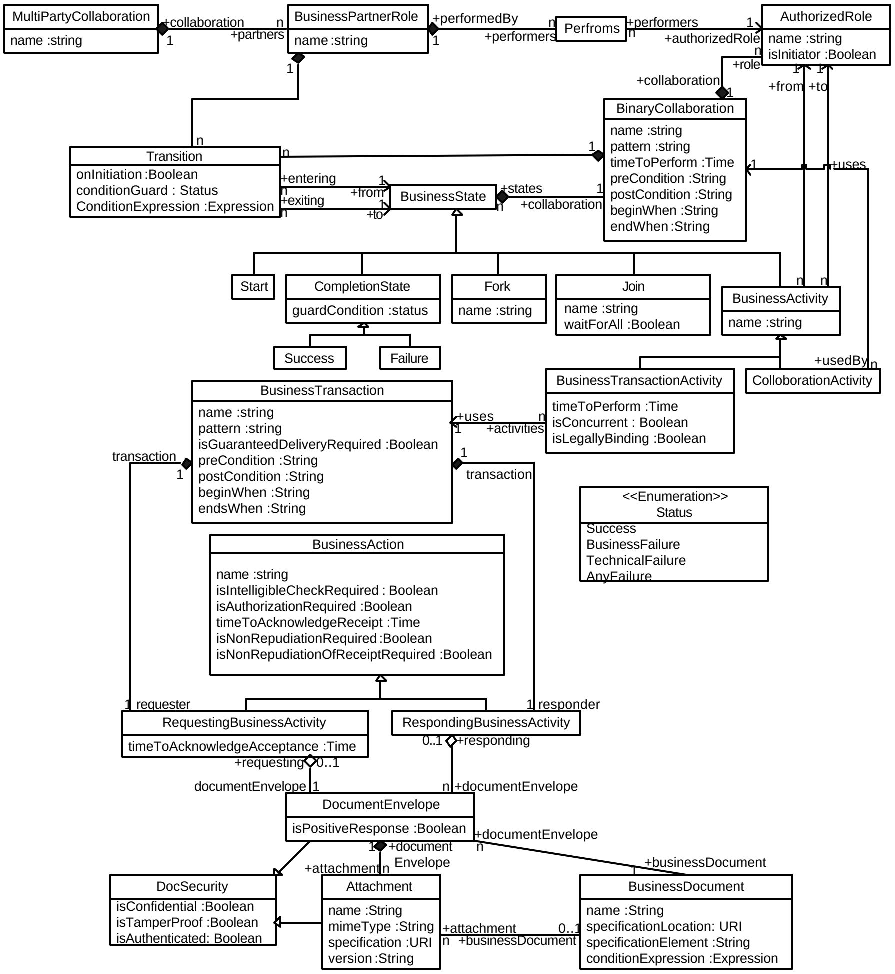

Fig. 4. ebXML Business Process Specification Schema, [**4**]

There are two kinds of Business Activities: Business Transaction Activities and Collaboration Activities. A Business Transaction Activity is the performance of a Business Transaction within the context of a Binary Collaboration. Thus, the same Business Transaction can be performed by multiple Business Transaction Activities within different Binary Collaborations or even within the same Binary Collaboration.

A Collaboration Activity is the performance of a Binary Collaboration. Analogous to Business Transaction Activities, a Binary Collaboration can be performed by multiple Collaboration Activities within different Binary Collaborations or even within the same Binary Collaboration.

A Binary Collaboration is not just an unordered set of Business Transaction Activities and Collaboration Activities. The Business Activities need to be ordered, which is done by means of a choreography. A choreography is specified in terms of Business States and Transitions between these states. The most important kind of a Business State is a Business Activity. Furthermore, there are a number of auxiliary Business States corresponding to diagramming artefacts on a UML activity chart: Start , Completion State, Fork , and Join.

5 Pragmatics of Business Actions

In the UMM Business Requirements View, there is only a very general relationship between economic concepts (i.e. Economic Event, Economic Contract, Economic Commitment, Economic Resource, Economic Resource Type, Agreement, and Partner Type) and process concepts. Essentially, the relationship states that a commitment or agreement is created by means of a collaboration, but there is no indication of how the constituents of the collaboration are related to the economic concepts. In order to get a more fine-grained view of the relationships between collaborations and economic concepts, we need to specify how the individual business actions involved in a collaboration are related to the economic concepts.

5.1 Pragmatic Actions

The basic notion introduced for relating business actions to economic concepts is that of a pragmatic action, see Fig. 5. A Pragmatic Action is a speech act, as defined in Section 2, and consists of two parts: a content and an illocutionary force. In e-commerce applications, the content is always an economic concept. The illocutionary force of a pragmatic action indicates in what way the action is related to its content. An agent can perform a pragmatic action and thereby influence an economic concept in a specific way.

Depending on which economic concept a pragmatic action addresses, different illocutionary forces are applicable. The pragmatic actions are, therefore, divided into several subclasses as indicated in Fig. 5. The three main subclasses are information actions, deontic actions, and fulfilment actions. The underling intuition for identifying these three sub classes of pragmatic actions is that in an e-commerce scenario, trading partners exchange business information, then establish different obligations, and finally exchange economic resources, thereby fulfilling the obligations.

An Information Action can have any economic concept as its content and requests or provides information about the concept. There are three possible illocutionary forces for information actions: Request asks for information, Reply answers a preceding request, and Provide provides information without a preceding request. Examples of information actions are “query for price and availability” and “query about status of order”.

A Fulfilment Action has an economic event as its content. The action may declare that an Economic Event has been performed, or it may express that such a declaration is accepted or rejected. There are three possible illocutionary forces for information actions: Declare states that an economic event has been performed, Accept states that a preceding declaration of performing an economic event is accepted, Reject states that such a declaration is rejected. Examples of fulfilment actions are “declare shipment completed” and “accept shipment”.

A Deontic Action can have a commitment or contract as its content. Thus, a deontic action concerns obligations to carry out events in the future. There are seven possible illocutionary forces for deontic actions: Propose means that an agent proposes the establishment of a commitment or contract. Accept is the acceptance of such a proposal while Reject is the rejection of a preceding proposal, Request-Cancellation is a request to cancel an established commitment or contract, AcceptCancellation is the acceptance of such a request, while RejectCancellation is the rejection of a preceding request to cancel, Cancel is a unilateral cancellation. Examples of Deontic Actions are “request purchase order” and “accept purchase order request”.

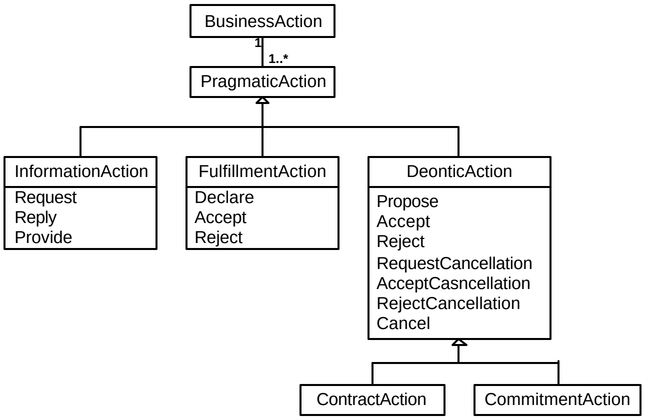

Fig. 5. Business Actions and Pragmatic Actions

5.2 Abstraction Levels of Business and Process Views

In the analysis of the relationships between the UMM/BRV and the BPSS process model (Fig. 2 and Fig. 4) it is important to recognise that parts of the corresponding concepts in the respective models are modelled on different levels of abstraction. Two common abstraction levels defined in [**12**] and [**5**], are the operational level and the knowledge level. The *operational level* models concrete, tangible individuals in a domain. The *knowledge level* models information structures that characterise categories of individuals at the operational level. Martin and Odell, [**14**], employ the concept of *power types* to refer to the correspondence between the objects of the knowledge and operational levels. A power type is a class whose instances are subtypes of another class. The Economic Resource Type of Fig. 2 is a power type of Economic Resource. Instances of the Economic Resource Type are the different categories of Economic Resources, for instance “real estate”. An instance of the Economic Resource class is a particular piece of land, e.g. “Hyde Park Mansions”.

In BPSS, classes like Business Activity and Business Transaction are defined at the knowledge level only. Business Activities do not possess properties related to actually transferred recourses, nor are they associated with the agents or roles between whom the transfer occurs. This is, however, not the case with the economic concepts of UMM/BRV. An Economic Event of UMM/BRV is explicitly related to an actual Economic Resource on the operational level. In other words, an Economic Resource refers to an actual and tangible resource, whereas an Economic Resource Type is the corresponding power type defined on the knowledge level, serving as a template for concrete Economic Resources. Pursuing this line of analysis it is possible to identify templates on several levels, each of which is the power type of the other.

To facilitate the integration with BPSS, whose constituents are modelled on the knowledge level only, several economic concepts need to be added to UMM/BRV to include classes defined on the knowledge level, which are presently not included. In the case of an Economic Contract, an Economic Contract Type is introduced to distinguish between the description of a contract and the actual contract between parties or abstract roles to be played by parties. The Economic Contract Type class models properties such as the types of conditions that may initiate or terminate a future contract, whereas an Economic Contract is associated to the authorised roles or partner types between whom a contract is established.

The introduction of new knowledge level classes into the economic concepts of UMM/BRV can be seen as schema conforming [**19**], i.e. transforming the schemas to be integrated in order to increase their similarity. The individual constituents of the BPSS model become possible to relate to the economic concepts of UMM/BRV as the two views now contain corresponding concepts defined on the same level of abstraction.

The global, integrated view of UMM/BRV and BPSS is shown graphically in Fig. 6. The glue in this integrated view are the pragmatic actions defined in section 5.1. Each individual Business Action of BPSS carries one or more Pragmatic Actions, which serve as categorisations of the Business Action. The categorisations are, in turn, defined in terms of the economic concepts of UMM/BRV. In Fig. 6, the original BPSS-parts are grouped with a dotted line boundary, UMM/BRV-parts are grouped with a dashed line boundary and the pragmatic actions that relate the two are depicted without any line boundary.

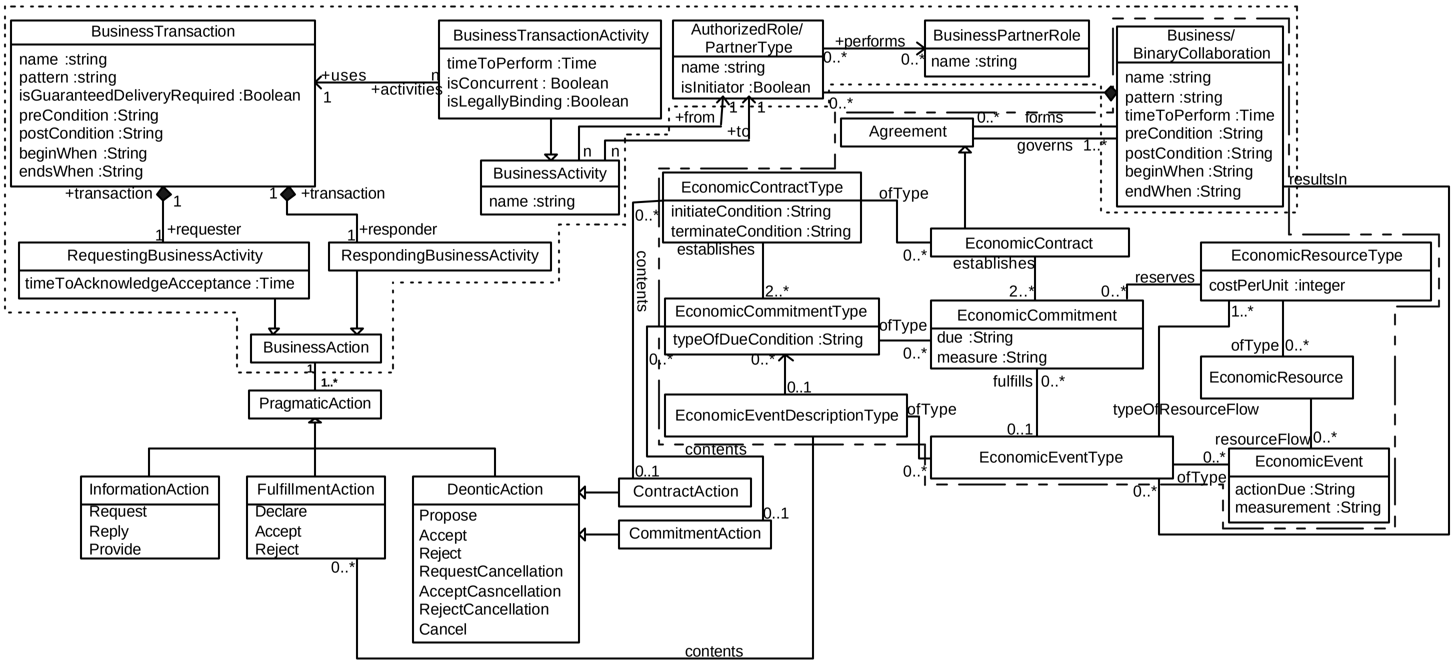
BusinessTransaction
name :string
pattern :string
isGuaranteedDeliveryRequired :Boolean
preCondition :String
postCondition :String
beginWhen :String
endsWhen :String
+transaction
1
+requester
1
RequestingBusinessActivity
timeToAcknowledgeAcceptance :Time
1
+transaction
1
+responder
RespondingBusinessActivity
BusinessAction
1
1..*
PragmaticAction
InformationAction
Request
Reply
Provide
FulfillmentAction
Declare
Accept
Reject
0..*
DeonticAction
Propose
Accept
Reject
RequestCancellation
AcceptCasncellation
RejectCancellation
Cancel
ContractAction
CommitmentAction
0..1
0..1
contents
contents
contents
BusinessTransactionActivity
timeToPerform :Time
isConcurrent : Boolean
isLegallyBinding :Boolean
+uses
1
n
+activities
BusinessActivity
name :string
n
n
+from
+to
1
1
AuthorizedRole/
PartnerType
name :string
isInitiator :Boolean
+performs
0..*
0..*
BusinessPartnerRole
name :string
0..*
Business/
BinaryCollaboration
name :string
pattern :string
timeToPerform :Time
preCondition :String
postCondition :String
beginWhen :String
endWhen :String
Agreement
0..*
forms
governs
1..*
resultsIn
EconomicContractType
initiateCondition :String
terminateCondition :String
establishes
0..*
ofType
EconomicContract
0..*
establishes
2..*
2..*
EconomicCommitmentType
typeOfDueCondition :String
ofType
0..*
0..*
0..*
EconomicCommitment
due :String
measure :String
0..*
reserves
fulfills
0..*
EconomicEventDescriptionType
0..1
ofType
0..1
EconomicEventType
0..*
EconomicResourceType
costPerUnit :integer
1..*
ofType
0..*
EconomicResource
typeOfResourceFlow
resourceFlow
0..*
0..*
EconomicEvent
actionDue :String
measurement :String
ofType
0..*

Fig. 6. Integrated view of business and process model

6 Applications

In this section, three applications of the introduced framework are introduced. First, we discuss how the framework can be used for understanding the semantics of UMM business transaction patterns. Secondly, different process views for incremental development are suggested. Finally, we outline a number of rules for governing the choreography of business collaborations.

6.1 Analysing and Extending UMM Business Transaction Patterns

UN/CEFACT has defined a number of business transaction patterns as part of UMM with the intention to provide an established semantics of frequently occurring business interactions. Below, we list and define a number of these patterns and show how they can be understood based on the framework introduced in the previous section.

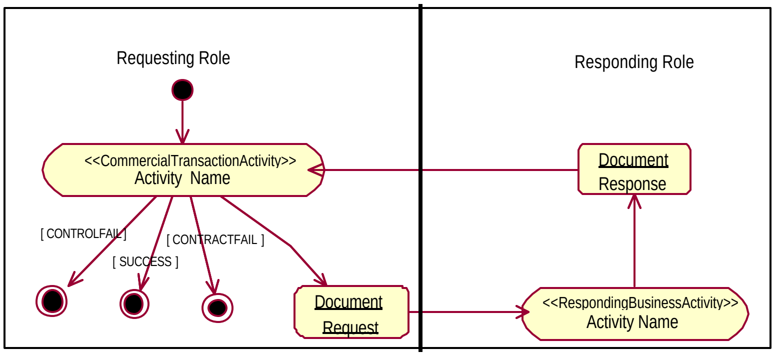

Fig. 7. Commercial Transaction Pattern

Commercial Transaction Pattern

Definition. "This design pattern is best used to model the '*offer and acceptance*' business transaction process that results in a residual obligation between both parties to fulfill the terms of the contract…The pattern specifies an originating business activity sending a business document to a responding business activity that may return a business signal or business document as the last responding message." [22] An activity diagram for this pattern is shown in Fig. 7; similar diagrams apply for the other patterns introduced below.

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Deontic Action and illocutionary force Propose. The pattern contains a Responding Business Activity with two pragmatic actions of type Deontic Action and illocutionary forces Accept and Reject, respectively.

Query/Response Pattern

Definition. “The query/response design pattern specifies a query for information that a responding partner already has e.g. against fixed data set that resides in a database. The response comprises zero or more results each of which meets the constraining criterion in the query.” [**22**]

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Information Action and illocutionary force Request. The pattern contains a Responding Business Activity with one pragmatic action of type Information Action and illocutionary force Reply.

Request/Response Pattern

Definition. “The request/response activity pattern shall be used for business contracts when an initiating partner requests information that a responding partner already has and when the request for business information requires a complex interdependent set of results.” [**22**]

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Information Action and illocutionary force Request. The pattern contains a Responding Business Activity with one pragmatic action of type Information Action and illocutionary force Reply. (Note that the analysis fails to make a distinction between the query/response and the request/response patterns; the reason for this is that the difference between the patterns does not reside in different business effects but in different ways of computing the responses.)

Request/Confirm Pattern

Definition. “The request/confirm activity pattern shall be used for business contracts when an initiating partner requests confirmation about their status with respect to previously established contracts or with respect to a responding partner’s business rules.” [**22**]

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Information Action and illocutionary force Request. The pattern contains a Responding Business Activity with one pragmatic action of type Information Action and illocutionary force Reply, and with the content Contract or Commitment.

Information Distribution Pattern

Definition. “This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt signal. The pattern is used to model an *informal* information exchange business transaction that therefore has no non-repudiation requirements.” [**22**]

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Information Action and illocutionary force Provide. The pattern contains a Responding Business Activity without any pragmatic action.

Notification Pattern

Definition. “This pattern specifies the exchange of a requesting business document and the return of an acknowledgement of receipt signal. The pattern is used to model a *formal* information exchange business transaction that therefore has non-repudiation requirements.” [**22**]

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Deontic Action and illocutionary force Declare. The pattern contains a Responding Business Activity without any pragmatic action. The motivation for this analysis is that a notification results in a binding specification of business conditions for the initiating partner and, thus, in a (partial) agreement.

The analysis suggests one way to interpret the definitions of the UMM transaction patterns, but it does not make any claims to be the final, “correct” interpretation of these definitions. This is not an achievable goal as the definitions are only formulated in natural language, sometimes quite vaguely. The value of the analysis is that it provides explicit interpretations that can be judged for their validity, and thereby can help in formulating more precise and unambiguous definitions of the patterns. Another use of the analysis is to suggest additional patterns. The following are obvious candidates for business transaction patterns:

Fulfilment Pattern

Definition. The fulfilment pattern specifies the completion of an economic event.

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Fulfilment Action and illocutionary force Declare. The pattern contains a Responding Business Activity with two pragmatic actions of type Fulfilment Action and illocutionary forces Accept and Reject, respectively.

Cancellation Pattern

Definition. The cancellation pattern specifies the cancellation of a contract or commitment.

Analysis. This pattern contains a Requesting Business Activity with one pragmatic action of type Fulfilment Action and illocutionary force RequestCancellation. The pattern contains a Responding Business Activity with two pragmatic actions of type Fulfilment Action and illocutionary forces AcceptCancellation and RejectCancellation, respectively.

6.2 Process Views

A process model may easily become complex and difficult to understand. One way to manage this complexity is to introduce a series of views of (partial) processes that move from the most basic actions to the finest details [**6**]. A series of such views is given below, based on the notions introduced in the previous section. Each view is an extension of the previous one and adds new components to the model. The background behind the order of the views is that the purpose of an e-Commerce process is to exchange economic resources. The first view, therefore, specifies the actual economic events taking place. The second and third views specify the contracts and commitments needed to manage the economic events. The following two views provides optional actions on information exchanges and cancellations.

As a working example, we will introduce a business model of an on-line e-Catering business, shown in Fig. 8. Here a customer places a meal order within a contract with an e-Caterer. Upon the receipt of a customer order, the e-Caterer purchases beverage from the beverage supplier and food from the food supplier, and packages it all into a customer meal delivery. The e-Caterer requests the customer to complete a down payment for her meal delivery and then completes a down payment for the beverage prior to the beverage delivery. Finally, the e-Caterer settles final payment for the beverage and payment for the food after receiving final payment from the customer for her meal delivery.

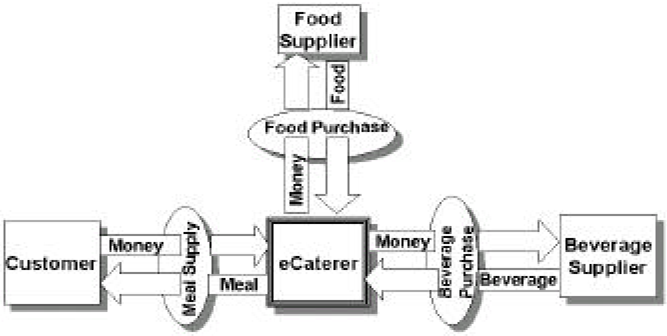

Fig. 8. The Business Model for an e-Caterer System

Fig. 9 – 13 show Activity Diagrams depicting the views of the transaction groups identified below. Fig. 14 shows the integrated view. Due to space limitations we have chosen to collapse every pair of Requesting and Responding Business Activity (see Fig. 7) into one BusinessTransaction. Thus the activity symbols of Fig. 9 - 14 refer to Business Transactions in the diagram and every symbol is placed on the swim lane between the Partner Roles (Customer, E-caterer etc.) to whom the Requesting and Responding Business Activities refers.

View 1. Fulfilment transactions

This view contains only Business Transactions of Business Actions related to Fulfilment Actions. These transactions represent the economic events taking place in the collaboration in which the transactions appear.

In the e-Catering case, a number of fulfilment transactions are depicted in Fig. 9.

View 2. Contract transactions

This view adds Business Transactions of Business Actions that are related to Contract Actions. These transactions represent the contracts needed for regulating the economic events of the collaboration.

A number of contract transactions from the e-Catering case are depicted in Fig. 10.(The transactions added are marked by a star.)

View 3. Commitment transactions

This view adds Business Transactions of Business Actions that are related to Commitment Actions. These transactions represent the detailed content of the contracts.

In the e-Catering case, a number of commitment transactions are shown in Fig. 11.

View 4. Information transactions

This view adds Business Transactions of Business Actions that are related to Information Actions. These transactions can, e.g., be about checking status of contracts and commitments.

In the e-Catering case, a number of information are depicted in Fig. 12.

View 5. Cancellation transactions

This view adds Business Transactions of Business Actions that are related to deontic actions concerning cancellations.

In the e-Catering case, a number of cancellation transactions are depicted in Fig. 13.

View 6. Choreography

The final process diagram consists of the completed choreography of the business collaboration, see Fig. 14.

The views introduced above can be used in several ways. First, they can be used in design. A designer could utilise the view guidelines by first constructing a process diagram according to view 1 and then gradually refine it until a set of diagrams in view 6 is obtained. Furthermore, the views can be used for presentation purposes. Business oriented users can choose to see only the top view or views, while designers and implementers can proceed to lower views. Even for the latter category of stakeholders, the layered views can help to understand a system by allowing to focus on an essential business perspective first and thereafter to proceed to a more detailed perspective.

6.3 Choreography Rules

In this section, we introduce three rules governing the choreography of business collaborations. Recall from section 4 that a *choreography* of a Business Collaboration is specified in terms of Business States. The relationships between these Business States are given by a directed graph, where an edge refers to a transition from one Business State to another. Furthermore, every Business State refers to exactly one Business Transaction, which means that we can restrict our attention to the choreography of Business Transactions without loss of generality.

When a designer constructs a choreography for a collaboration, it is helpful to consider the dependencies that exist among the transactions of the collaboration. There exist two kinds of dependencies that occur across many domains: trust dependencies [**11**] and flow dependencies [**13**].

A *trust dependency* is an ordered pair of transactions <A, B>, which expresses that A has to be performed before B as a consequence of limited trust between the initiator and responder. As an example, it is possible to require that a product be paid before it can be delivered.

A *flow dependency* is an ordered pair of transactions <A, B>, which expresses that A has to be performed before B because the economic resources obtained in A are needed for carrying out B. As an example, the different components of a meal to be delivered must be transferred to the e-Caterer before she can deliver the complete meal to the customer.

We define two partial orders, Flow and Trust, whose members are flow and trust dependencies, respectively.

Trust is a partial order over $(Ful \cup Com \cup Ctr) \times (Ful \cup Com \cup Ctr)$.
Flow is a partial order over $Ful \times Ful$.

Ful, *Com*, *Ctr* and *Can* refer to the sets of Fulfillment, Commitment, Contract, and Cancellation transactions, defined in views 1 – 5 above, respectively.

The following rule can now be stated based on flow and trust dependencies:

Rule1: If A and B are nodes in a choreography **C**, and $\langle A,B \rangle \in \{Flow \cup Trust\}$ then there must exist a path from A to B in **C**.

Furthermore, we observe that the establishment of a commitment or contract must precede the cancellation of the same, which gives rise to the following rule:

Rule 2: If A and B are nodes in a choreography **C** and $A \in \{Com \cup Ctr\}$ and $B \in Can$ where B is cancelling the contract or commitment established by A then there must exist a path from A to B in **C**.

Returning to the relationships between Economic Commitment, Economic Contract and Economic Event, we observe that Economic Contracts are subtypes of Agreements carrying Economic Commitments that some actual economic exchange will be fulfilled in the future. Thus we identify the following rule:

Rule 3: If A and B are nodes in a choreography **C** and $A \in \{Com \cup Ctr\}$ and $B \in Ful$, where B is establishing the economic event that fulfils the commitment established by A, then there must exist a path from A to B in **C**.

Rules 1 - 3 can be used to guide and restrict the design of a choreography, i.e. give suggestions for possible paths between different transactions and rule out incorrect paths.

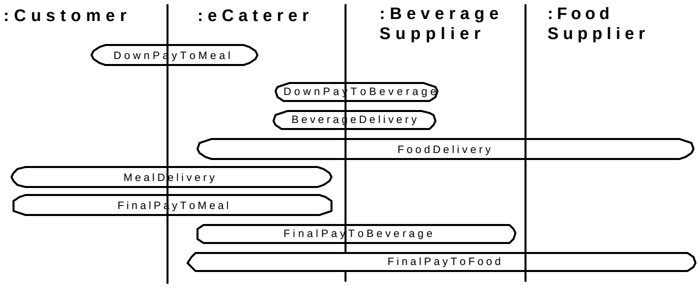

Fig. 9. View 1 – Fulfillments

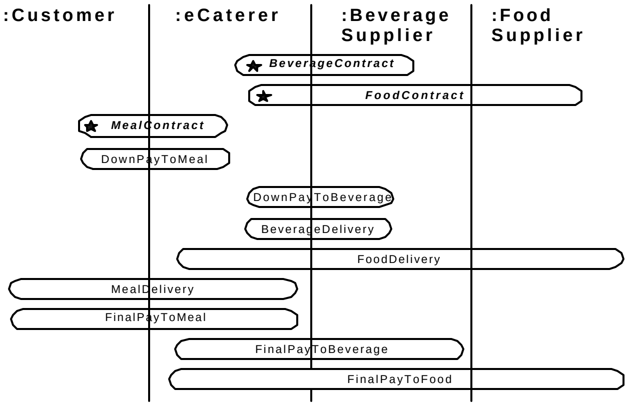

Fig. 10. View 2 – Contracts

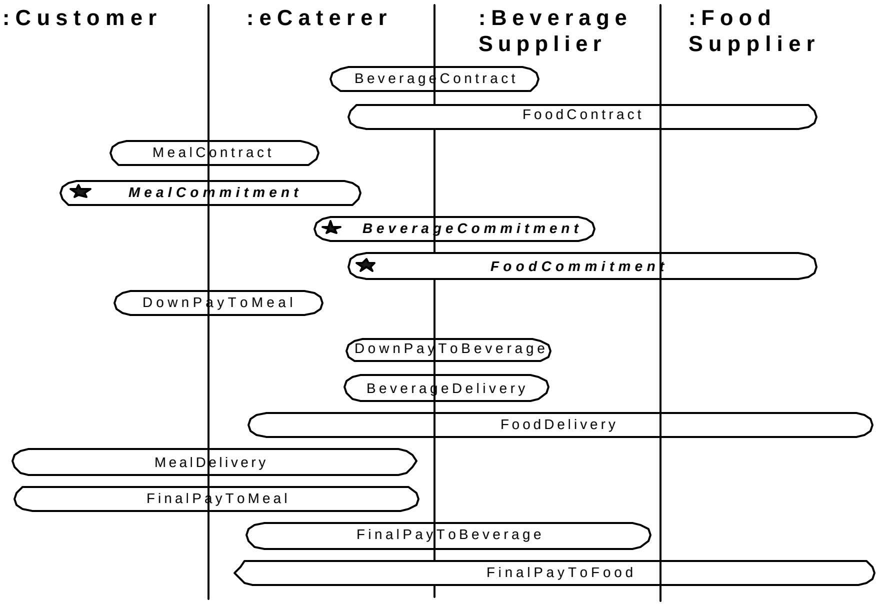

Fig. 11. View 3 – Commitments

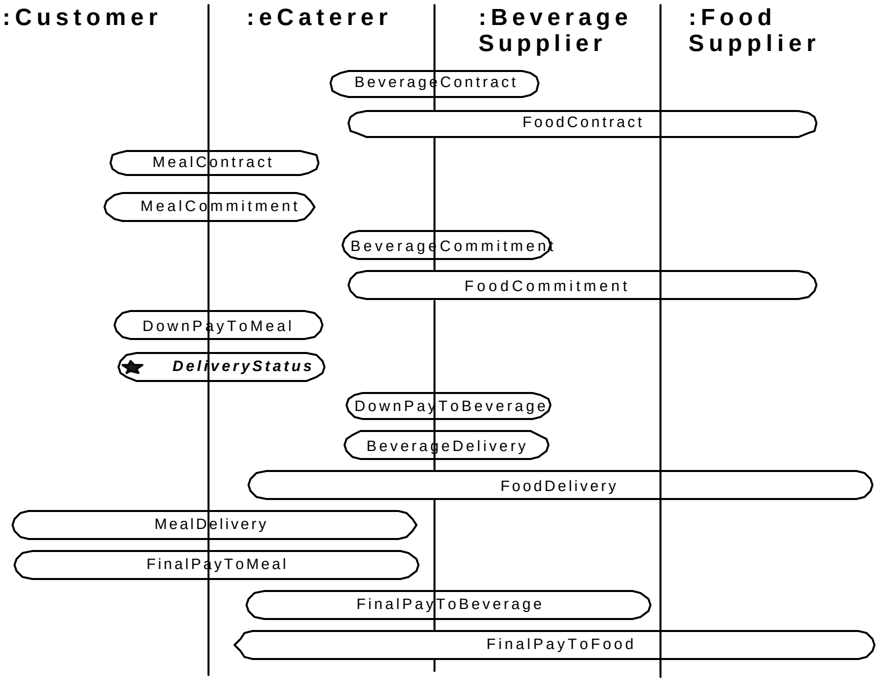

Fig. 12. View 4 – Information

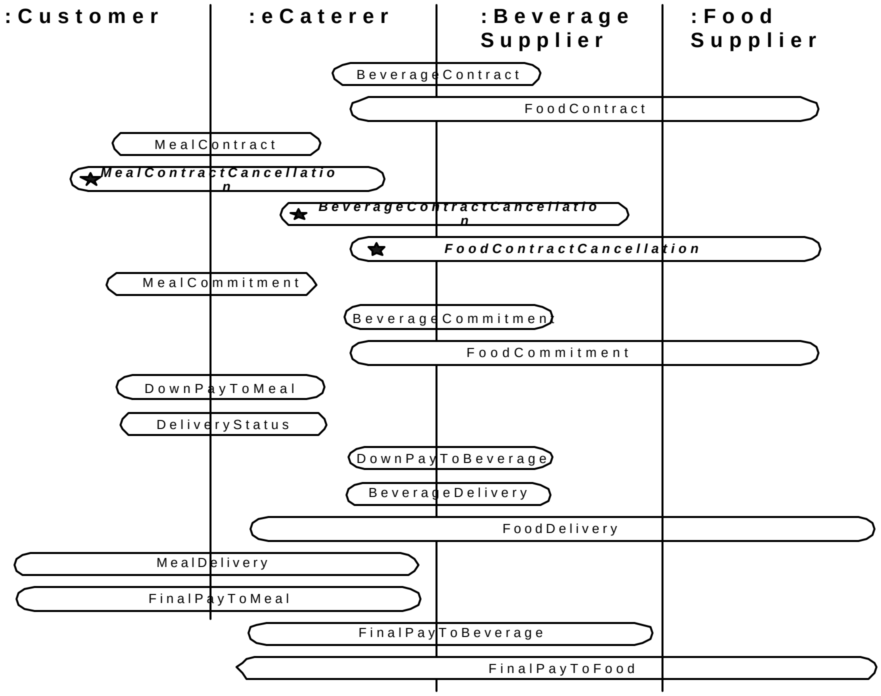

Fig. 13 View 5 – Cancellation

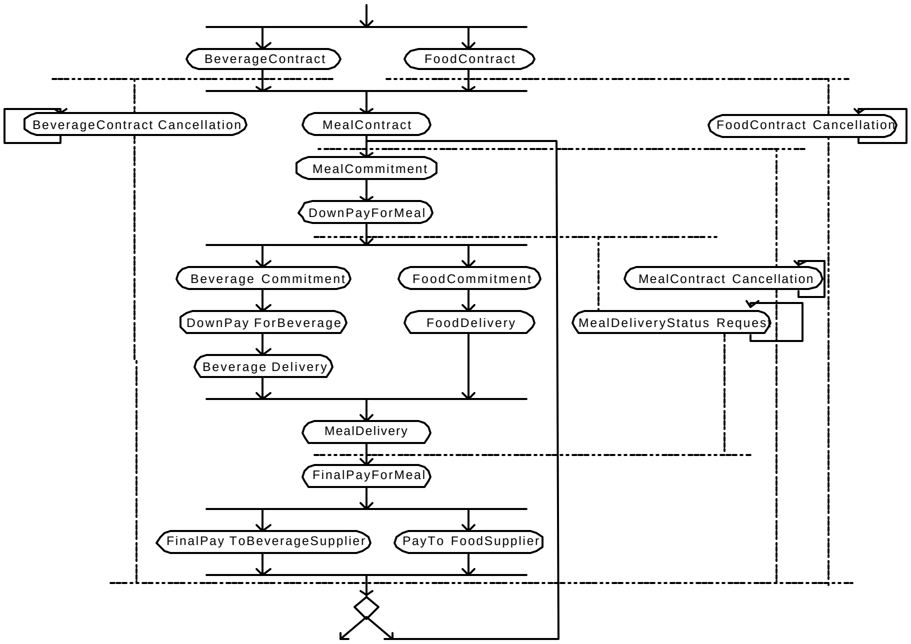

Fig. 14 View 6 – Choreography

7 Concluding Remarks

The main contribution of this paper is a unified framework to facilitate the analysis and integration of business models and process models in e-Commerce. The approach suggested bridges the gap between the declarative aspects of a business model and the procedural aspects of a process model. The work has been carried out and expressed in the context of the ebXML standard, but the results can easily be adapted to other frameworks.

Recent approaches to e-Commerce systems design, [**8**], [**3**], stress the distinction between business models and process models. They also suggest that business models should be developed independently of process models. This is a valid design approach as it separates concerns, i.e. separates reasoning on declarative economic aspects from procedural control flow aspects. However, these aspects have to come together in the final design. The framework proposed in this paper can help in merging business and process aspects in a systematic way.

Another area where the proposed framework can be of use is process integration. Classifying concepts of process models in terms of their pragmatic content, can in this respect be used in several ways: as a measurement of the degree of similarity in the establishment of correspondences between models, to support the mapping of contexts and concepts between different e-Commerce standards, and also aid conflict analysis, i.e. offer ways of categorising conflicts that may occur due to introduced mappings between the models to be integrated.

The work in this paper focuses on integration processes, i.e. processes for exchanging resources across the borders of organisations. An interesting extension is to consider also internal processes that describe the work going on within an organisation. This would mean addressing aspects such as delegation, authorities, and accounting responsibilities, [**9**].

References

1. Austin J. L., "*How to do things with Words*", Cambridge, MA: Harvard University Press, 1962.
2. Dietz, J.L.G., and Mallens P.J.M., "Business Process Modeling as a Starting Point for Information Systems Design", *Data2Knowledge Newsletter, Part 1 (january 2001), 2 (march 2001) and 3 (may 2001).*
3. ebXML Business Process and Business Information Analysis Worksheets and Guidelines version 1.01, http://www.ebxml.org/specs/bpWS.doc
4. ebXML Business Process Specification Schema v1.01 (XML schema and DTD examples available separately) http://www.ebxml.org/specs/ebBPSS_print.pdf (Valid on March 08th, 2002).
5. Geerts G. and McCarthy W. E., "The Ontological Foundaton of REA Enterprise Systems", *working paper, Michigan State University* (August 2000 and being revised for journal submission).
6. Johannesson P. and Perjons E., "Design Principle for Application Integration", *12th Conference on Advanced Information Systems Engineering*, eds. B. Wangler and L. Bergman, Springer LNCS, 2000.

7. Goldkuhl G., “Generic business frameworks and action modelling”, In proceedings of *Conference Language/Action Perspective*, Springer Verlag, 1996.
8. Gordijn J., Akkermans J. M. and Vliet J. C., “Business Modeling, is not Process Modeling”, *eCOM2000 workshop, 19th International Conference on Conceptual Modeling*, 2000.
9. Weigand H., de Moor A., “A Framework for the Normative Analysis of Workflow Loops”, *ACM SIGGROUP Bulletin*, August 2001/Vol 22, No.2
10. http://www.ebxml.org
11. Jayaweera P., Johannesson P. and Wohed P., “Process Patterns to Generate eCommerce Systems”, *2nd International Workshop on Conceptual Modeling Approaches for e-Business*, to be held in conjunction with the 20th International Conference on Conceptual Modeling (ER2001) Yokohama, Japan.
12. Fowler M., “*Analysis Patterns: Reusable Object Models*”, Addison-Wesley, 1997
13. Malone et al.: “Towards a handbook of organizational processes”, *MIT eBusiness Process Handbook*, http://ccs.mit.edu/21c/mgtsci/index.htm.
14. Martin, J., Odell, J.: *Object-Oriented Methods. A Foundation,* Prentice Hall 1994
15. McCarthy W. E., “The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment”, *The Accounting Review* 1982.
16. Medina-Mora R. et al.: “The Action Workflow Approach to Workflow Management Technology”, *Proceedings of 4th Conference on Computer Supported Cooperative Work*, ACM Press, 1992.
17. Object Constartint Language Specification, Version 1.1, 01 September 19997, http://www.omg.org/docs/ad/97-08-08.pdf, (Valid on March 08th, 2002).
18. OMG Unified Modeling Language Specification, Version 1.4, September 2001, ftp://ftp.omg.org/pub/docs/formal/01-09-67.pdf, (Valid on March 08th, 2002).
19. Spaccapietra S., Parent C. and Dupont Y., “Model Independent Assertions for Integration of Heterogeneous Schemas”, *The VLDB Journal*, vol. 1, no.2, pp. 81-126, 1992.
20. Searle J. R., “A taxonomy of illocutionary acts”, K. Gunderson (Ed.), *Language, Mind and Knowledge*, Minneapolis: University of Minnesota, 1975.
21. Significant *REA Model* papers: http://www.msu.edu/user/mccarth4/paplist1.html, (Valid on March 08th, 2002).
22. TMWG N090R10: UN/CEFACT Modeling Methodology (UMM). N090, http://www.ebxml.org/project_teams/jdt/resources/, (Valid on March 08th, 2002).