

# Process Patterns to Generate e-Commerce Systems

Prasad Jayaweera, Paul Johannesson, and Petia Wohed

Department of Computer and Systems Sciences  
Stockholm University and Royal Institute of Technology  
Electrum 230, SE-164 40 Kista, Sweden  
{prasad, pajo, petia}@dsv.su.se

**Abstract:** In electronic commerce, two fundamental types of models are business models and process models. While a business model is concerned with value exchanges between actors, a process model describes the procedural realization of business requirements. There is a need for methodological guidelines and tool support to move from a business model to a process model, which enables design decisions to be based on requirements captured in the business model. This paper addresses a systematic transformation of business models to process models. We propose a designer assistant that systematically aids a designer in generating a process model in an executable process modeling language.

## 1 Introduction

When building e-commerce systems, two types of models are fundamental: business models and process models, [4]. The purpose of a business model is to describe the fundamental business aspects of the e-commerce system to be built. Thus, a business model describes which actors are involved, what the actors offer each other, and what activities they perform when producing and consuming offerings. The central concept in a business model is that of *value*, and the model describes how value is exchanged between actors [9]. This can be contrasted to a process model, which aims at describing the operational and procedural aspects of a process and specifies the control flow of the activities carried out in a process. A process model specifies the actors involved in the operations, which activities they perform as well as the sequencing of these activities. Thus, a business model defines the *what* in an e-commerce system, while a process model defines the *how*.

A business model can be seen as more basic than a process model as it specifies the declarative aspects of an e-commerce system. A natural question is, therefore, whether it is possible to move from a business model to a process model in a systematic way. Methodological support for this task would provide several benefits: support for identifying design alternatives in process modeling, support for motivating design choices, and a clarification of the relationships between the declarative business model and the procedural process model.

In this paper, we argue that it is possible to systematically move from a business model to a process model, and we suggest methodological guidelines and modeling

techniques that can assist a designer in the task. A starting point of the method proposed is that much of the procedural aspects of a process model concern communication among actors. This communication is carried out in order to establish commitments among the actors to perform exchanges of values. The commitments are created by speech acts, and the control flow in a process is determined by the interleaving of these speech acts with each other and with the value exchanging activities.

The proposed guidelines and techniques are based on four building blocks:

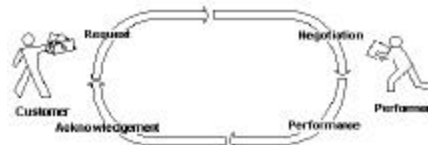
1. Business Model
2. BML, a formal and executable process definition language
3. Predefined Process Patterns
4. Automated Designer Assistant

The paper is an extension of process patterns in our work [6] and is structured as follows. Section 2 outlines relevant research in the area and introduces our conceptual framework. Section 3 introduces the identified process patterns built on communicative acts. Section 4 describes the automated designer assistant that supports the task of creating process models. Section 5 concludes the paper and suggests directions for further research.

## 2 Related Research and Conceptual Framework

The approach proposed in this paper is a combination of elements from the Language Action approaches to information systems design, which focus on communication aspects, and elements from the work by Gordijn et. al. [4], who focus on the transfer of values between actors and explicitly distinguish between business and process models.

Language Action approaches to information systems design build on Austin's Speech Act Theory [1], which acknowledges speech acts as a special action category. A speech act is defined as an action (changing the universe of discourse) performed by a speaker and grasped by a recipient. Searle [12] further develops the theory by introducing a taxonomy of five different kinds of speech acts, namely: *assertive*, *directive*, *commissive*, *expressive*, and *declarative*. Well-known Language Action approaches are Action Workflow Loop [10], Business Action Theory (BAT) [3], and Dynamic Essential Modeling of Organization (DEMO) [2]. We have adopted Action Workflow Loop (**Fig. 1**) for the definition of proposed process patterns.



**Fig. 1.** Action Workflow Loop with four phases

The conceptual framework we use is based on UMM's economic model describing resources, events and agents (REA model) [15], [5]. In order to make the model

suitable also for communication aspects, we have extended it by concepts from speech act theory. Furthermore, we include a number of notions proposed by Weigand et. al. [16] used for distinguishing between different levels of communication.

The extended REA model (**Fig. 2**) describes business events and business entities as well as their structure and relationships. The basic business entities are Agreement, Economic Resource Type and Economic Resource. An Agreement is an arrangement between two Partner Types. Furthermore, an Agreement with any economic commitment is called Economic Contract and specifies the economic exchanges to occur, which is represented by the entity Commitment. An example of an Economic Contract is a purchase order where each order line is a Commitment. A Commitment is an obligation from one Partner Type to another to transfer the specified Economic Resource Type, and it is fulfilled through an Economic Event.

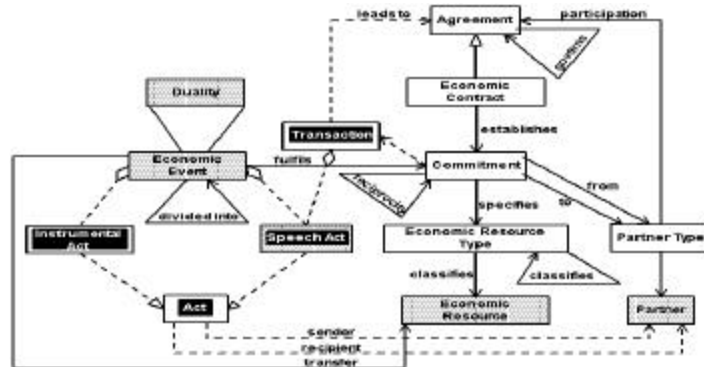
The most central concepts in our approach [the shaded rectangles in (**Fig. 2**)] are the following:

**Partner:** A partner is an independent economic and/or legal entity, e.g., John Doe, Stockholm University.

**Economic resource:** An economic resource is a quantity of something of value that is under the control of a partner, e.g., a car.

**Economic event:** An economic event is the transfer of the control of an economic resource from one partner to another partner, e.g., the ownership change of a car.

**Duality:** The corresponding term used by Gordijn, [4], is “value offering” and is used to represent the relationship between two economic events, T1 and T2, that satisfy the condition: if T1 is an economic event transferring an economic resource from partner A1 to partner A2, then T2 is the corresponding economic event transferring an economic resource from A2 to A1. The intuition is that the Duality represents the reciprocity between the economic events - one partner providing another partner with something of value and receiving something of value in return. An example of Duality is a car purchase where a car ownership is transferred from a retailer to a customer and the corresponding payment is provided.



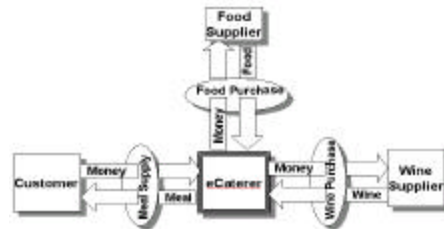
**Fig. 2.** An extended UMM economic model over REA  
 (The black labels, as well as the dotted relationships, mark the extensions of the original model.  
 The shaded rectangles represent the central concepts in our approach)

We also allow an economic event to be **divided into** several Economic Events. This may occur when a monetary transfer is done, e.g., a payment is divided into a down payment and a final payment, or when a delivery of a large order is divided into several shipments. Furthermore, applying speech act theory, we claim that an economic event consists of one or more **Speech Act(s)** and, optionally, an Instrumental Act. An Instrumental Act represents the physical transfer of an Economic Resource from one Partner to another one, whereas Speech Act models the communication between the partners. Finally, applying Weigand's levels for communication patterns [16], some pairs of speech acts build Transactions, e.g., a request/commit. Some Transactions lead to Agreements, while others directly result in Commitments.

Now, we define a **Business Model** as a triple  $\langle PT, ER, DT \rangle$ , where

- PT - a set of partner types
- ER - a set of economic resource types
- DT - a set of duality types

An example of a business model is shown in **(Fig. 3)**. This model represents four partner types: eCaterer, Customer, WineSupplier, and FoodSupplier. The customer orders meals from the eCaterer who purchases wine from the WineSupplier and food from the FoodSupplier and prepares the meal. In this example,  $PT = \{Customer, eCaterer, WineSupplier, FoodSupplier\}$ ,  $ER = \{Meal, Money, Wine, Food\}$ ,  $DT = \{\langle\langle eCaterer, Customer, Meal \rangle, \langle Customer, eCaterer, Money \rangle\rangle, \langle\langle WineSupplier, eCaterer, Wine \rangle, \langle eCaterer, WineSupplier, Money \rangle\rangle, \langle\langle FoodSupplier, eCaterer, Food \rangle, \langle eCaterer, FoodSupplier, Money \rangle\rangle\}$ . DT represents the dualities: MealSupply, FoodPurchase, and WinePurchase.



**Fig. 3.** Business Model for eCatering Example

The process definition language BML (has similarities to SDL [11]) is the formal and executable language based on communicating state machines used in our work. Interested readers are referred to [7]. BML diagrams are designed from one partner's (called here, the base partner) perspective and model messages sent from and received by the base partner.

In our work we have used BML symbols: *convex and concave hexagon* for send and receive message respectively, *diamond* for business decision, *ellipses* for start, wait and stop states, and finally *stick man* for a partner involved (**Fig. 5**), (**Fig. 6**). We have extended, the original BML syntax by allowing wait states to receive messages and also to send message out from *Stop* state, simply for the compactness and clarity of process model but preserving a straight forward mapping between the original syntax and our extended version.

### 3 Generic Process Patterns

In this section, we introduce and describe four patterns for e-commerce in the form of BML diagrams. We hypothesize that most process models for ecommerce can be expressed as a combination of these four diagrams.

- *Economic Resource Requesting Diagram (ERRD)*. This diagram is an action-workflow loop from the perspective of the requesting actor, i.e. when the requesting actor is the base partner. For example, a base partner's purchase order enacts one ERRD instance to acquire supplier delivery.
- *Economic Resource Offering Diagram (EROD)*. This diagram is an action-workflow loop from the perspective of the supplying actor, i.e. when the supplying actor is the base partner. For example, a customer's order enacts one EROD instance for the fulfillment of customer delivery.
- *Providability Checking Diagram (PCD)*. This diagram models the reservation and booking of resources needed for carrying out an economic event.
- *Counter Offer Diagram (COD)*. This diagram handles communication carried out in order to identify alternatives to a request that could not be fulfilled.

These diagrams and their intercommunication are shown in (Fig. 4). A typical scenario would be the following: a customer orders an economic resource by initiating an EROD instance. The EROD checks the availability of required resources by starting a PCD instance, which communicates with a supplier first directly and then through an ERRD instance. PCD first reserves and then books the necessary economic resources. Reservation, holding resources for a very short time, is weaker compared to booking, which leads to ordering [7]. If the request by the customer cannot be fulfilled, counter offer management can be initiated through a COD instance.

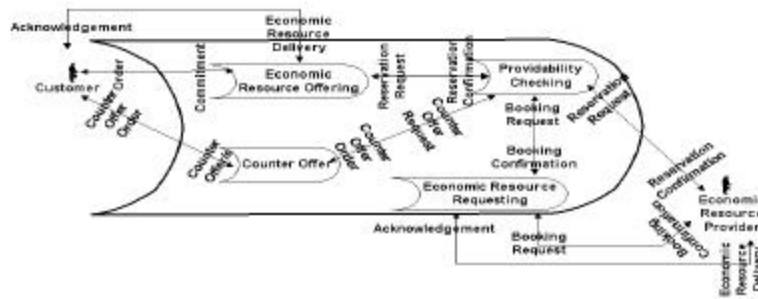


Fig. 4. BML Static Diagram for Generic four Process Patterns

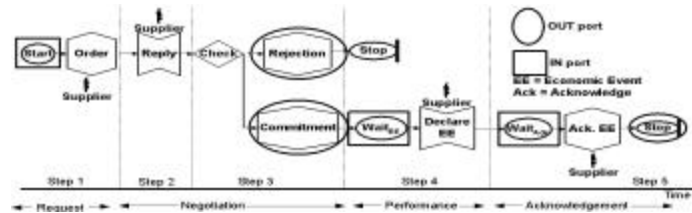
#### 3.1 Economic Resource Requesting Diagram (ERRD)

An Economic Resource Requesting Diagram (Fig. 5) models a situation where the base partner receives an economic resource from another actor, called supplier.

**Step 1:** The base partner sends an order to the supplier requesting an economic resource.

**Step 2:** The supplier sends a reply for the order made by the base partner.

- Step 3:** The supplier's reply is interpreted either as a rejection or as a commitment to fulfil the order.
- Step 4:** The supplier declares the delivery of the ordered economic resource.
- Step 5:** The base partner acknowledges the receipt of the requested economic resource.



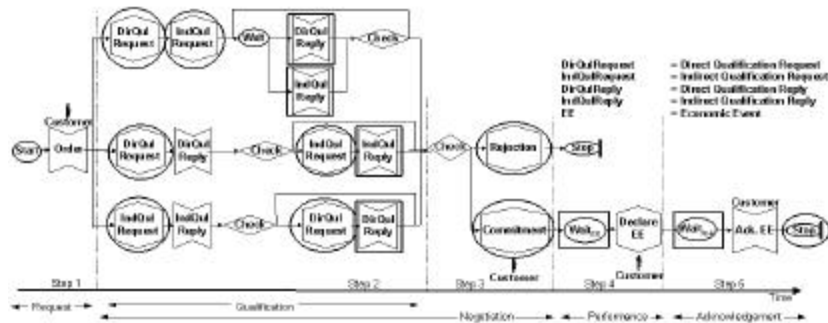
**Fig. 5.** Economic Resource Requesting Diagram (ERRD)

In this basic BML diagram, there are some explicit positions where inter diagram communication is possible. Sending out positions are called *OUT ports* while receiving in positions are called *IN ports*.

### 3.2 Economic Resource Offering Diagram (EROD)

An Economic Resource Offering Diagram models a situation where the base partner supplies another actor, called customer, with an economic resource (**Fig. 6**). In this case, we follow the suggestion by James Taylor in [13] and introduce additional qualification steps, where the base partner acquires *direct* and *indirect* means required for carrying out the requested action. Direct means are resources from suppliers needed to carry out the requested action. Indirect means are resources from the customer corresponding to the duality of the economic event.

- Step 1:** The customer sends an order for an economic resource to the base partner.
- Step 2:** The qualification steps are handled by choosing one of three alternatives. In the first one, there is no order between acquiring direct and indirect means. In the second one, direct means are acquired before indirect means. In the third one, indirect means are acquired before direct means. This includes one or more synchronize request-reply to acquire all necessary means prior to commit recipient with a delivery.
- Step 3:** The replies received when acquiring direct means are evaluated, and the base partner either rejects the customer's order or commits to deliver the customer order.
- Step 4:** The base partner declares the delivery of the ordered economic resource to the customer.
- Step 5:** The customer acknowledges the receipt of the ordered economic resource.



**Fig. 6.** Economic Resource Offering Diagram (EROD)

All the steps prior to sending the message “Commitment” of ERRD and EROD can be viewed as a commitment establishment process between the involved partners, while the succeeding steps can be viewed as a commitment fulfillment process.

Note that there is an asymmetry between the economic resource requesting and offering diagram. The reason for this is that the qualification steps are relevant only when the base partner has to supply an economic resource. The additional qualification steps may lead to two other fundamental process patterns Providability Checking and Counter Offer that are described below. In the case of an economic resource requested by the base partner, these two process patterns are not under her control.

### 3.3 Providability Checking Diagram (PCD)

The providability checking diagram (Fig. 7) models a situation where direct means for an economic resource offering is first reserved and then booked. In the preliminary reservation, PCD communicates directly with a supplier. When the booking request is received from EROD, it invokes an ERRD to make the formal ordering. The Providability Checking Diagram (PCD) may invoke counter offer handling when the base partner is not capable of providing the original order.

- Step 1:** A reservation request is received from the economic resource offering diagram (EROD).
- Step 2:** The reservation request is sent to a supplier and a reservation reply is received.
- Step 3:** The reservation reply is evaluated either to a counter offer exception, which is to be handled by a Counter Offer Diagram (COD) for the economic resource, or to a reservation confirmation that is to be sent to an EROD.
- Step 4:** A booking request from EROD is sent to an ERRD.
- Step 5:** When the booking confirmation from the ERRD is received, EROD is acknowledged with the received booking confirmation.

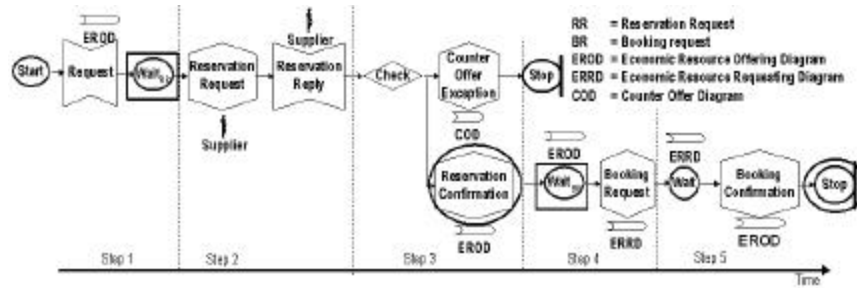


Fig. 7. Providability Checking Diagram with marked IN and OUT ports (PCD)

### 3.4 Counter Offer Diagram (COD)

A Counter Offer Diagram (Fig. 8) models the management of counter offers, which are received from a PCD. A COD communicates with a Counter Offer Provider and with a customer in order to identify an alternative for the original request made by the customer.

- Step 1:** A counter offer request is received from PCD.
- Step 2:** A counter offer request is sent to the Counter Offer Provider (COP) and the reply is received. (A COP is a process that creates counter offers; it may use advanced algorithms and/or include human involvement.)
- Step 3:** The response received from COP is evaluated either to a rejection that is to be notified to the customer or to a set of counter offers that the customer may choose among.
- Step 4:** The customer can reply with a rejection, request a selected offer, or request an alternative offer.
- Step 5:** If the customer rejects the counter offer, the rejection is directed to a relevant handler process here called Rejection Recorder. If the customer requests a selected offer, then the request is sent to new PCD instance. If the customer requests an alternative, the request is sent to a new COD instance.

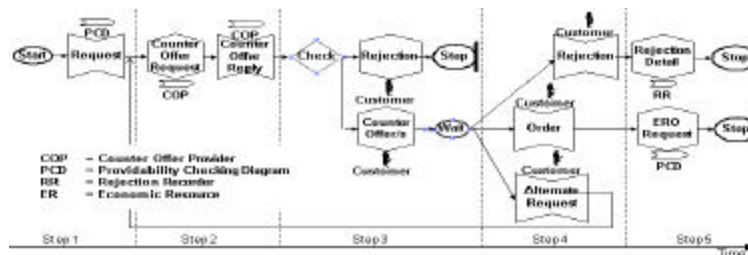


Fig. 8. Counter Offer Diagram with explicit interactions (COD)

These four basic process patterns can be group into direct process patterns and intermediate process patterns. The first two patterns (ERRD and EROD) fall into



direct process patterns where initial interaction is with external partner and also notice that there are no inter-diagram communications identified at the definition level. The last two patterns (PCD and COD) fall into intermediate process patterns where enactment is caused by one instance of the direct process pattern. Therefore at the definition level, some inter-diagram communications have marked of intermediate process patterns.

### 3.5. Process Model

A process can be modeled by a set of Economic Resource Requesting Diagrams, Economic Resource Offering Diagrams, optionally Providability Checking and Counter Offer Diagrams – such a set is called a *process model*. The basic structure of the diagrams in a process model can be derived simply from a business model. However, the communication among the diagrams is not uniquely determined by the business model, but may vary depending on the requirements for the process. How to determine this communication, i.e. how to move from a business model to a process model is the main topic of the next section.

## 4 A Designer Assistant

In this section, we will show how a business model can be transformed and extended into a process model in a systematic way. Moving from a business model to a process model is not a trivial task but requires a large number of design decisions. In order to support a designer in this task, we propose an automated designer assistant that guides the designer through the task by means of a sequence of questions (a similar designer assistant has been proposed for the area of conceptual modeling by Wohed [17]). The questions can be divided into four phases, (Fig. 9).

**Phase 1.** The designer builds the business model, identifies the base partner, i.e. the organisation from whose perspective the system is to be built, and the customer of the process.

**Phase 2.** The designer establishes a (partial) order between the economic events of the business model.

**Phase 3.** The designer introduces the communicative acts needed to handle the economic events and establishes a (partial) order between them.

**Phase 4.** From the output of phase 3, a process model is automatically derived.

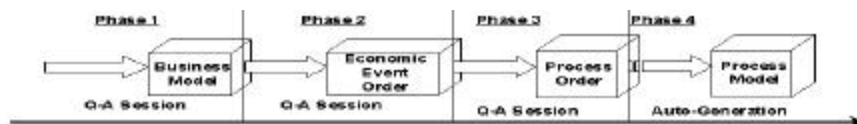


Fig. 9. Four phased approach to process models

#### 4.1 Phase 1 - Business Model

In the first phase, the designer builds a business model and specifies the organization for which the e-commerce system is to be developed, as well as its customer. Due to the space limitations, this phase is omitted and interested readers are referred to [6]. However, in order to clarify, we exemplify the result from this phase by referring back to (Fig. 3) where the business model for the described example is shown. We will further use this example as a running example through the rest of the paper.

#### 4.2 Phase 2 - Economic Event Order (EEO)

In this phase, the designer starts to construct an order between the activities of the process. First, the designer takes into account only economic events while disregarding the communicative acts that co-ordinate the process. By considering only the order of the economic events in this phase, the designer can concentrate on the main business logic and postpone until later more detailed design decisions about the coordination of communicative acts.

The designer first has to decide whether an economic event must or can be divided into parts; such a part is called an *atomic economic event*. The first question is therefore:

1. *What are the atomic economic events of each economic event?*

After having identified and named the atomic economic events, the designer is prompted to order them by determining the dependencies that exist between them. In an e-commerce context, we identify two main types of dependencies: trust dependencies and flow dependencies.

A *trust dependency* occurs between two atomic economic events within the same duality, e.g. that a product must be paid before it can be delivered. A trust dependency expresses the level of trust between the actors involved in a duality, e.g. requesting a down payment expresses low trust. A *flow dependency*, [8], occurs between two atomic economic events in different dualities and expresses that the economic resource obtained by one of the economic events is needed for the other economic event. A simple example is that a retailer has to obtain a product from an importer before delivering it to a customer. In order to identify trust and flow dependencies, the following question is posed.

2. *How do you order the atomic economic events?*

An example of answers to the questions 1 and 2 is given in (Fig. 10).

1. What are the Atomic Economic Events (AEE) of each Economic Event (EE)?

Economic Event	Atomic Economic Event	Abbreviation
<Caterer, Customer, Meal>		MealToCustomer (MTC)
<Customer, eCaterer, Money>	<Customer, eCaterer, DownPay>	DownPayFromCustomer (DPFC)
	<Customer, eCaterer, FinalPay>	FinalPayFromCustomer (FPFC)
<WineSupplier, eCaterer, Wine>		WineToCateringCompany (WTCC)
<Caterer, WineSupplier, Money>	<Caterer, WineSupplier, DownPay>	DownPayToWineSupplier (DPTWS)
	<Caterer, WineSupplier, FinalPay>	FinalPayToWineSupplier (FPTWS)
<FoodSupplier, eCaterer, Food>		FoodToCateringCompany (FTCC)
<Caterer, FoodSupplier, Money>		FinalPayToFoodSupplier (FPTFS)

2. How do you order the Economic Events and the Atomic Economic Events?  
(Add only = symbol in the matrix)

	MTC	DPFC	FPFC	WTCC	DPTWS	FPTWS	FTCC	FPTFS
MealToCustomer (MTC)			=					
DownPayFromCustomer (DPFC)	=							
FinalPayFromCustomer (FPFC)								
WineToCateringCompany (WTCC)	=							
DownPayToWineSupplier (DPTWS)				=				
FinalPayToWineSupplier (FPTWS)								
FoodToCateringCompany (FTCC)	<							<
FinalPayToFoodSupplier (FPTFS)								

The table shall be read in the following way: row header, cell, column header, e.g. the first cell on the seventh row gives FoodToCateringCompany precedes MealToCustomer.

Fig. 10. Questions and answers for the example in phase 2

#### 4.2.1 Reservation and Booking

When the offered economic resource is assembled from more than one economic resources, the acquisition of direct means for each can be ordered by getting answers for the questions below. The economic event order can be completed with reservation and booking as follows. An example answer to the questions is given in (Fig. 11).

- 3a. Do you reserve economic resource  $ER_1$  before reserving economic resource,  $ER_2$ ?
- 3b. Do you book economic resource  $ER_1$  before booking economic resource,  $ER_2$ ?
- 3c. Do you book economic resource  $ER_1$  before reserving economic resource  $ER_2$ ?

The resulting partial order (EEO) from phase 2 is the following (“<” means precedes):

EEO = {MealToCustomer < FinalPayFromCustomer, DownPayFromCustomer < MealToCustomer, DownPayFromCustomer < FinalPayFromCustomer, WineToCateringCompany < MealToCustomer, WineToCateringCompany < FinalPayToWineSupplier, DownPayToWineSupplier < WineToCateringCompany, DownPayToWineSupplier < FinalPayToWineSupplier, FoodToCateringCompany < MealToCustomer, FoodToCateringCompany < FinalPayToFoodSupplier, BookWine < BookFood, ReserveWine < ReserveFood}

Such a partial order between atomic economic events is called an *economic event order*. It expresses the order between the most important activities in the process and disregards coordinative activities.

3. For each Atomic Economic Event, identified the ones needed with counter offer handling.

- Wine Supply with counter offer handling
- Food Supply with counter offer handling

Column	Wine	Food
Row		
Wine		3a. BookWine < BookFood? Yes 3b. ReserveWine < ReserveFood? Yes 3c. BookWine = ReserveFood? No
Food	3a. BookFood < BookWine? - 3b. ReserveFood < ReserveWine? - 3c. BookFood = ReserveWine? -	

i.e. Get answers for three questions:  
Book(row) = Book(column)?  
Reserve(row) = Reserve(column)?  
Book(row) = Reserve(column)?  
considering a pair at a time.

Fig. 11. Questions and Answers for Reservation and Booking

### 4.3 Phase 3 - Process Order

In phase 3, the designer will extend the economic event order from phase 2 by specifying dependencies between communicative acts. A starting point for this task is that for each atomic economic event, there will be one action-workflow loop. The designer has to determine the interactions between the loops given by all the atomic economic events. Like earlier, the designer assistant supports this task through a number of questions. The intuition behind several of these questions is, roughly expressed, the following: Before an actor does something of value to another actor, it will check whether that actor has deserved it. By doing “something of value to another actor” is meant to carry out an economic event, to commit to carry out an economic event, or to initiate the acquisition of means needed to carry out an economic event. The expression “check whether that actor has deserved it” has to do with the fact that an economic event from an actor A to an actor B always is accompanied by another economic event from B to A; recall that these two economic events together constitute one duality. The expression states that before actor A is prepared to carry out its economic event (or some preparation to it) to B, it will check whether B has done its corresponding economic event (or some preparation). Note that this check will be done only if the economic event order so prescribes. Furthermore, there are questions for ensuring that all required means for carrying out an economic event have been obtained.

In order to formulate the questions, we need to distinguish between an *incoming atomic economic event* (AEE), where the base partner receives an economic resource, and an *outgoing AEE*, where the base partner supplies an economic resource.

If In is an incoming AEE and Out an outgoing AEE within the same duality, and In < Out in the economic event order, ask:

- 4a. Do you require that In be performed before you commit to perform Out?
- 4b. Do you require a commitment for In before you commit to perform Out?

Furthermore, if In is an incoming AEE and Out an outgoing AEE within the same duality, and Means is an incoming AEE in another duality, and In < Out, and Means < Out in the economic event order, ask:

- 5a. Do you require that Means be performed before you commit to perform Out?
- 5b. Do you require a commitment for Means before you commit to perform Out?
- 5c. Do you require that In be performed before you request Means?

An example of answers to these questions is given in (Fig. 12). All answers will result in an extension to the economic event order from phase 2, which also includes ordering between communicative acts. Such an order is called a *process order*. In this case, we arrive at a process order PO:

```
PO = EEO U
{decl(DownPayFromCustomer) < com(MealToCustomer), com(WineToCateringCompany) < com(FinalPayToWineSupplier),
com(FoodToCateringCompany) < com(FinalPayToFoodSupplier), com(WineToCateringCompany) < com(MealToCustomer),
decl(DownPayFromCustomer) < dir(WineToCateringCompany), com(FoodToCateringCompany) < com(MealToCustomer),
decl(DownPayFromCustomer) < dir(FoodToCateringCompany)}
```

EEO is the economic event order derived in phase 2 for the example. Abbreviations for communicative acts used in PO are: ‘*dir*’ for directive, ‘*com*’ for commissive, and ‘*decl*’ for declarative.

4a. Do you require *DownPayFromCustomer* be performed before you commit to perform *MealToCustomer* (Yes/No)? Yes  
 It is unnecessary to ask 9b as the answer to 9a already implies it.

4a. Do you require *WineToCateringCompany* be performed before you commit to perform *FinalPayToWineSupplier* (Yes/No)? No

4b. Do you require a commitment for *WineToCateringCompany* before you commit to perform *FinalPayToWineSupplier* (Yes/No)? Yes

4a. Do you require *FoodToCateringCompany* be performed before you commit to perform *FinalPayToFoodSupplier* (Yes/No)? No

4b. Do you require a commitment for *FoodToCateringCompany* before you commit to perform *FinalPayToFoodSupplier* (Yes/No)? No

5a. Do you require that *WineToCateringCompany* be performed before you commit to perform *MealToCustomer* (Yes/No)? No

5b. Do you require a commitment for *WineToCateringCompany* before you commit to perform *MealToCustomer* (Yes/No)? Yes

5c. Do you require *DownPaymentFromCustomer* be performed before you request *WineToCateringCompany* (Yes/No)? Yes

5a. Do you require that *FoodToCateringCompany* be performed before you commit to perform *MealToCustomer* (Yes/No)? No

5b. Do you require a commitment for *FoodToCateringCompany* before you commit to perform *MealToCustomer* (Yes/No)? Yes

5c. Do you require *DownPaymentFromCustomer* be performed before you request *FoodToCateringCompany* (Yes/No)? Yes

Fig. 12. Example of answers to question in section 4.3

#### 4.4 Phase 4 - Mapping Process Order to BML Process Model

The basis in generation of a process model is that for each atomic economic event, there is an action workflow loop represented in a process model by one of the direct process patterns, i.e. an ERRD or an EROD. The process order, identified through the second and the third step is necessary for identifying the inter-diagram connections as well as whether it is necessary to include an intermediate process pattern, i.e., a PCD or a COD or not. Basically, each inequality identifies one inter-diagram connection. The formal rules for this mapping are, due to space limitation, omitted here, but may be found in [14].

However, to give a basic intuition of the logic behind them, we are exemplifying the effect of a couple of the mapping rules. For example an inequality telling that the wine has to be delivered to the catering company before the catering company can in turn deliver the meal to the customer ( $WineToCateringCompany < MealToCustomer$ ), should result in a connection from the stop state of the ERRD modeling the caterer's wine request process, to the wait for confirmation state in the EROD modeling the caterer's meal offering process (Fig. 13). This inter-diagram connection indicates that the wine delivery to catering company process has to be completed before the meal delivery process can go on, which indeed is the semantic of the inequality.

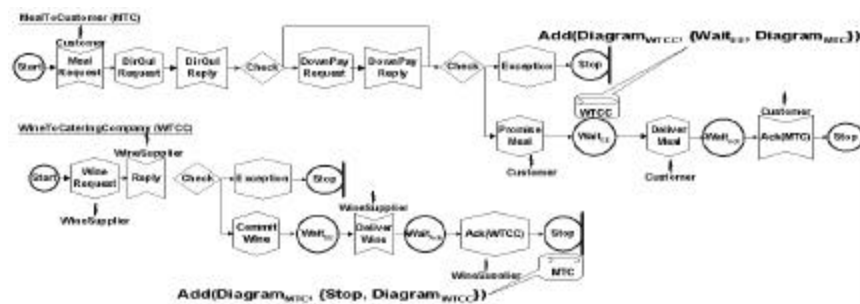


Fig. 13. BML process segment (MealToCustomer and WineToCateringCompany processes)

Another example of an inequality telling that getting down payment from customer before ordering wine from wine supplier [ $decl(DownPayFromCustomer) <$

dir(WineToCateringCompany)], will connect stop state of ERRD modeling down payment from customer and start state of ERRD modeling wine to catering company, indicating that completion of former enacts the latter process instance (Fig. 14).

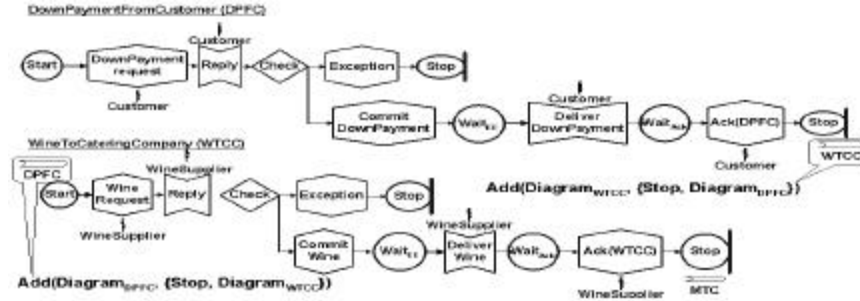


Fig. 14. BML process segment (DownPayFromCustomer and WineToCateringCompany processes)

The relevant two rules that complete inter process communication for the above two inequalities can be listed as below.

<p>If <math>AEE_i &lt; AEE_j</math> then  Add(DIAGRAM<sub>i</sub>, {Stop, DIAGRAM<sub>j</sub>})  Add(DIAGRAM<sub>i</sub>, {Wait<sub>ij</sub>, DIAGRAM<sub>j</sub>})</p>	<p>If <math>dec(AEE_i) &lt; dir(AEE_j)</math> then  Add(DIAGRAM<sub>j</sub>, {Stop, DIAGRAM<sub>i</sub>})  Add(DIAGRAM<sub>j</sub>, {Start, DIAGRAM<sub>i</sub>})</p>
---	---

## 5 Concluding Remarks

The main contribution of this paper is a set of methodological guidelines that support a designer in moving from a business model to a process model in a systematic way. The approach provides a number of advantages:

**Identifying alternatives.** The designer assistant helps the designer to identify and evaluate possible design alternatives when building the process model and thereby ensures that no useful alternatives are overlooked.

**Traceability and motivation.** When inspecting a process model, it is often difficult to understand why a particular solution has been chosen. By building a process model by means of the designer assistant, all design choices as well as their motivations are automatically and explicitly recorded.

**Separation of concerns.** The approach suggested makes an explicit distinction between the declarative aspects of a business model and the procedural aspects of a process model. This separation of concerns aids a designer in focussing on one problem at a time.

**Seamless transition from analysis to realization** Using the designer assistant, the designer starts with a business model and builds successively a process model on its basis. The end point of this activity is a set of diagrams that can be used for communication about the model as well as for actual execution.

In this paper, we have only covered the simplest form of an e-commerce process. Further work is, therefore, needed to handle extensions such as negotiations,

breakdowns, cancellations, etc. Furthermore, the scope of the processes could also be extended to handle additional phases in e-commerce, like contact search as in BAT.

## 6 References

1. Austin J.L., "*How to do things with words*", Oxford, 1962.
2. Dietz, J.L.G.; Barjis, J.: "Petri Net expressions of DEMO Process Models as a rigid foundation for Requirements Engineering", *The 2nd Int. Conference on Enterprise Information Systems (ICEIS'00)*, 2000.
3. Goldkuhl G (1998) "The six phases of business processes - business communication and the exchange of value", *12<sup>th</sup> biennial ITS conference (ITS '98)*, Stockholm.
4. Gordijn J., Akkerman s J. M. & Vliet J. C.: "What's in an Electronic Business Model? ", *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th Int. Conference*, Springer-Verlag 2000.
5. Greets G. and McCarthy W. E., "*The Ontological Foundation of REA Enterprise Information Systems*", Michigan State University (revised for journal submission), 2000.
6. Jayaweera P., Johannesson P., & Wohed P.: "From Business Model to Process Patterns in e-commerce", *6<sup>th</sup> Int. Workshop on the Language-Action Perspective on Communication Modeling* 2001, Montreal, Canada.
7. Johannesson P. and Perjons E.: "Design Principle for Application Integration", *12th Conference on Advanced Information Systems Engineering*, Springer LNCS, 2000.
8. Malone et al.: "Towards a handbook of organizational processes", *MIT eBusiness Process Handbook* <http://ccs.mit.edu/21c/mgtsci/index.htm>, 08<sup>th</sup> of Nov. 2001.
9. Porter M. E.: "*Competitive Advantages. Creating and Sustaining Superior Performance*" The Free Press 1998.
10. Raul Medina et al.: "The Action Workflow Approach to Workflow Management Technology", *4<sup>th</sup> Conference on Computer Supported Cooperative Work*, 1992.
11. SDL Standards, <http://www.sdl-forum.org/Publications/Standards.htm>, 08<sup>th</sup> of Nov. 2001.
12. Searle J.: "*Speech Acts - An Essay in the Philosophy of Language*", Cambridge University Press 1969.
13. Taylor J.: "The Limits of Rationality in Communication Modeling – a Semiotic Reinterpretation of the Concept of "Speech Act"", *3<sup>rd</sup> Int. Workshop, The Language Action Perspective on Communication Modeling*, 1998.
14. Technical Report, "Mapping Function to Generate BML Process Model", url: <http://www.dsv.su.se/~prasad/html/MapFun.doc>, 08<sup>th</sup> of Nov. 2001.
15. UN/CEFACT, url: <http://www.unece.org/cefact/>, 08<sup>th</sup> of Nov. 2001.
16. Weigand H., Moor A. de & Heuvel W-J. van den, "Supporting the evolution of workflow patterns for virtual communities", *33<sup>rd</sup> Hawaii Int. Conference of System Sciences*, 2000.
17. Wohed, P., "Conceptual Patterns for Reuse in Information Systems Analysis", *12th Int. Conference on Advanced Information Systems - CAiSE 2000*.