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 generates 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, [5]. 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 [12]. 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 modelling, 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. A business model describing the values exchanged in an e-commerce process.
2. A formal and executable language based on communicating state machines used for modeling processes.
3. A number of predefined process patterns built on communicative acts.
4. An automated designer assistant that guides a user from a business model to an executable process model.

The paper is an extension of our work in [7] and structured as follows. Section 2 introduces our conceptual framework and outlines relevant research in the area. Section 3 describes BML, a formal language for process modeling. Section 4 introduces the identified process patterns built on communicative acts. Section 5 describes the automated designer assistant that supports the task of creating process models. Section 6 concludes the paper and suggests directions for further research.

2 Conceptual Framework and Related Research

The approach proposed in this paper is a combination of elements from the Language Action approaches to information systems design [8], 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. The classical example for a speech act is the utterance made by a priest during a wedding ceremony "I hereby, pronounce you husband and wife". Searle [14] further develops the theory by introducing a taxonomy of five different kinds of speech acts, namely: *assertive*, a speech act which conveys information about some state of affairs from the speaker to the hearer; *directive*, a speech act where the speaker requests the hearer to carry out some action; *commissive*, the speaker commits himself to carry out some action; *expressive*, the speaker expresses his attitude about some state of affairs; and *declarative*, the speaker brings about some state of affairs.

One of the most well known Language/Action approaches is the Action Workflow approach, [22] and [10]. In this approach, business processes are modelled as loops, see **Fig. 1**~~**Fig. 1**~~. A loop starts by a request from a customer, followed by a negotiation phase, which results in the provider accepting the request and promising to carry it out. The third step consists of the provider carrying out the request, and the last step is the acknowledgement of the customer that the request has been satisfied.

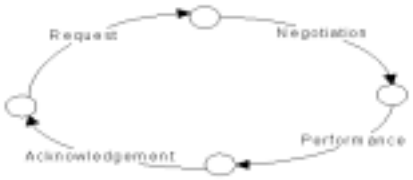


Fig. 1. Action Workflow Loop with four phases

Another Language Action approach is BAT (Business Action Theory), [3]. BAT has a more limited scope than the Action Workflow approach as it only addresses business transactions and not work processes in general. For business processes, BAT provides a more elaborated framework for business transactions than Action Workflow by also incorporating preliminary phases, such as contact search. An important feature of BAT is the symmetry it introduces by stating that both actors in a business transaction have mutual obligations to each other. This idea is also supported by Gordijn [4], and is a central element of the approach presented here.

In contrast to the Language Action modeling, the approach we propose starts with a Business Model of reciprocal economic events. Based on this business model and a number of pre-identified process patterns grounded on communicative acts, a final executable process model is deduced with the help of a designer assistant.

The conceptual framework we use is based on UMM's economic model describing resources, events and agents (REA model) [18]. 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 at al [20], [21] used for distinguishing between different levels of communication.

The REA model, Fig. 2~~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 one to transfer the specified Economic Resource Type, and it is fulfilled through an Economic Event.

The most central concepts in our approach (the coloured rectangles in Fig. 2~~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 exchange" 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.

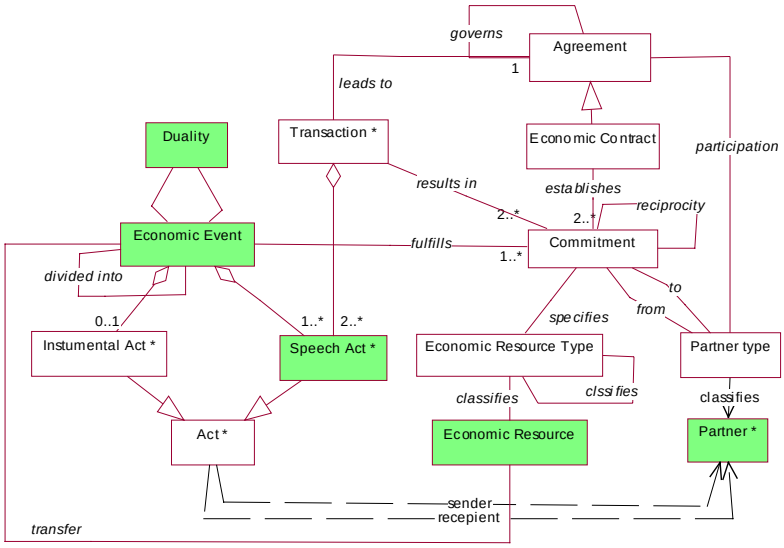


Fig. 22. An extended UMM economic model over resources, events and agents
(The asterisk, as well as the dotted relationships, mark the extensions of the original model. The coloured rectangles represent the central concepts in our approach)

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 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 [20], some couples 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 <PT,ER,DT>, 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~~**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 = {<<eCaterer, Customer, Meal>, <Customer, eCaterer, Money>>, <<WineSupplier, eCaterer, Wine>, <eCaterer, WineSupplier, Money>>,<<FoodSupplier, eCaterer, Food>, <eCaterer, FoodSupplier, Money>>} DT represents the dualities: MealSupply, FoodPurchase, and WinePurchase.

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

3 Business Modeling Language

This section briefly introduces a language based on communicating state machines, BML (Business Modeling Language), which is developed by Viewlocity, [19], [9]. The language has similarities to SDL (Specification and Description Language), [2], [13]. BML is a communication oriented process language, which means that it focuses on describing interaction between actors through the sending and receiving of messages. An important advantage of BML is that it can be used for the specification and design as well as maintenance of systems. This means that the same language can be used in different phases of a system's life cycle: in feasibility analysis, in requirement specification, in the design and implementation phases, and even in the operation phase. This enables different categories of stakeholders to use the same language for different purposes.

BML can describe the structure as well as the behaviour of a system by using two kinds of graphical diagrams. The structure of the system is visualised by a static

diagram, see Fig. 6~~Fig. 6~~, which describes the processes in a static mode. The dynamic behavior of a system is described by using process models, which visualize the order in which the messages shall be sent and received, see Fig. 4~~Fig. 4~~.

Fig. 4. The BML diagram visualizes the order in which the messages shall be sent and received in a process. *Note that the figure only shows the beginning of a Process.*

The process segment shown in Fig. 4~~Fig. 4~~ describes the situation when Message 1 is received from Process A, Message 2 is sent out to the Customer. Then it waits for Message 3 from Process B.

The main BML symbols are the following, see also Fig. 5~~Fig. 5~~:

Receive Message: describes the consumption of a message from the input queue.
Send Message: describes the sending of a message.
Automated Business Decision: The control flow is dynamically changed depending on different business rules.
Actor: symbols of external actors. (An actor is a wider concept including a partner, an artificial agent, or any other external agent able to communicate with the system.)
Wait for Event and **Start:** The process instance is waiting in the Wait for Event state until a message is received. A Wait for Event symbol with a name "Start" is the starting state.
Stop: describes the end of the flow of the process instance.

Fig. 5~~5~~. Symbols used in BML

For our work we have made two extensions to the original BML semantics, mainly to ensure the compactness and the clarity of targeting process models. First, receive and send messages can be received from or be sent to more than one process or actor, secondly, wait states also can receive message/s prior to making the transition to the next state.

A basic characteristic of a BML diagram is that it is designed from one partner's perspective; and we will refer to him as a *base partner*. The base partner sends messages to, and receives messages from other actors. Typically, the base partner is the organization for which an e-commerce system is to be built.

4 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 e-commerce 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.
- *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.
- *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. 6~~Fig. 6~~**. 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 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 [21~~21~~]. If the request by the customer cannot be fulfilled, counter offer management can be initiated through a COD instance.

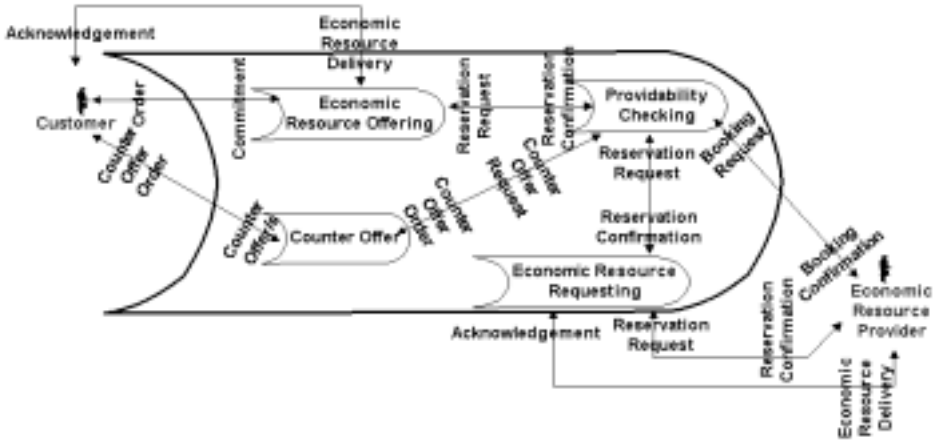


Fig. 6~~6~~. BML Static Diagram for Generic four Process Patterns

4.1 Economic Resource Requesting Diagram

An Economic Resource Requesting Diagram (Fig. 7~~Fig. 7~~) 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.

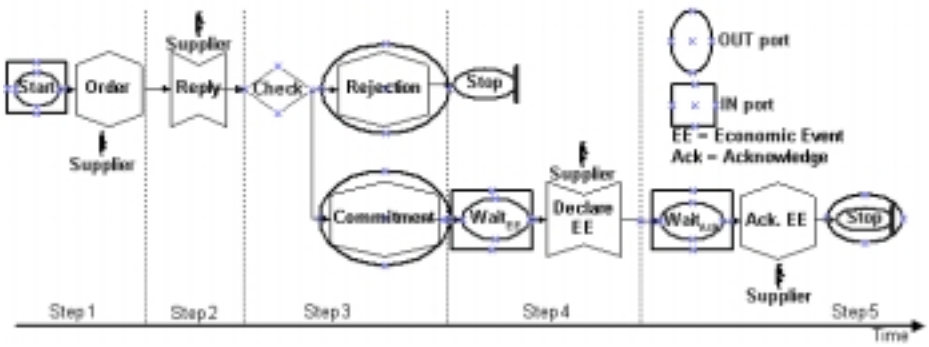


Fig. 7~~7~~**.** (ERRD) Economic Resource Requesting Diagram

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*.

4.2 Economic Resource Offering Diagram

An Economic Resource Offering Diagram models a situation where the base partner supplies another actor, called customer, with an economic resource (see Fig. 8~~Fig. 8~~). In this case, we follow the suggestion by James Taylor in [16] 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 delivering an economic resource.

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 ordered economic resource.

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.

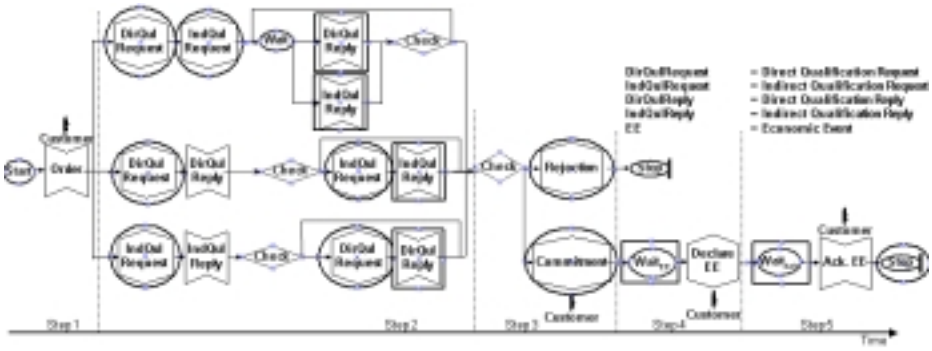


Fig. ~~8~~8. (EROD) Economic Resource Offering Diagram

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 which are described below. In the case of an economic resource requested by the base partner, these two process patterns are not under her control.

4.3 Providability Checking Diagram

The providability checking diagram (**Fig. 9~~Fig. 9~~**) 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.

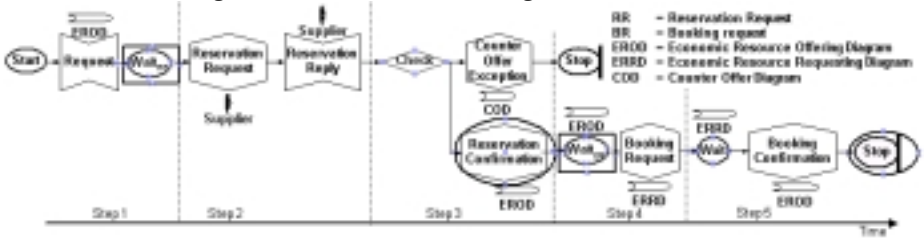


Fig. ~~9~~9. (PCD) Providability Checking Diagram with marked IN and OUT ports

4.4 Counter Offer Diagram

A Counter Offer Diagram (COD, see **Fig. 10~~Fig. 10~~**) 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 even 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.

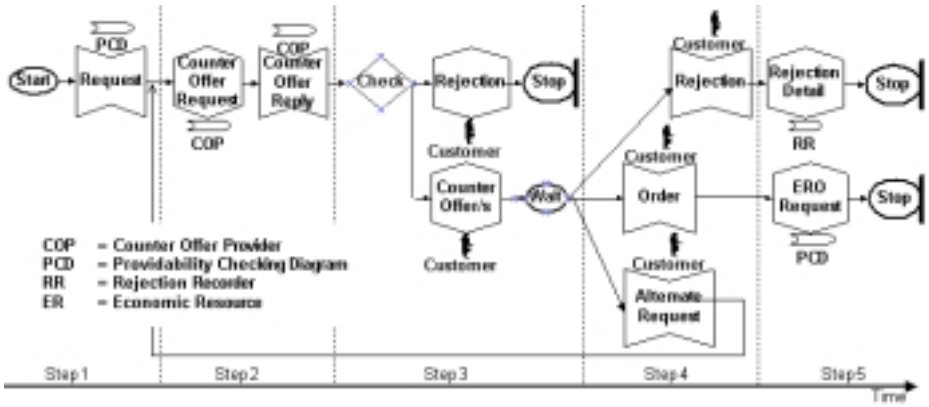


Fig. 10~~10~~. (COD) Counter Offer Diagram with explicit interactions.

4.5. 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.

5 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. A business model, as defined in Section 3, states what economic resources are exchanged between what actors, while a process model, as defined in Section 4, shows the order of the actors' activities in the form of communicative acts. 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 [], [].). The questions can be divided into four phases, see **Fig. 11~~Fig. 11~~**.

Phase 1. The designer builds the business model, identifies the base partner, i.e. the organization 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.

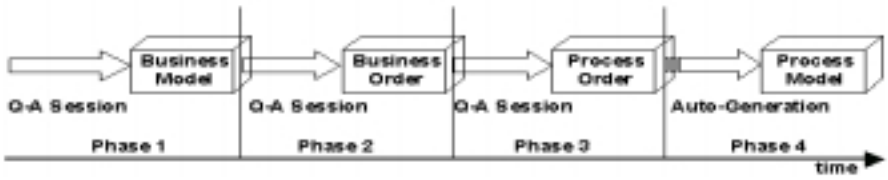


Fig. 11~~11~~. Four phased approach to process models.

5.1 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 [7]. However, in order to clarify, we exemplify the result from this phase by refering back to

Fig. 3~~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.

5.2 Phase 2 - Business Order

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:

7. 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*, [11], 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.*8. How do you order the atomic economic events?*

An example of answers to the questions 7 and 8 is given in **Fig. 12**~~**Fig. 12**~~.

7. What are the Atomic Economic Events (AEEs) of each Economic Event (EE)?

Economic Event	Atomic Economic Events	Abbreviation
<eCaterer, Customer, Meal>		MealToCustomer
<Customer, eCaterer, Money>	<Customer, eCaterer, DownPay>	DownPayFromCustomer
	<Customer, eCaterer, FinalPay>	FinalPayFromCustomer
<WineSupplier, eCaterer, Wine>		WineToCateringCompany
<eCaterer, WineSupplier, Money>	<eCaterer, WineSupplier, DownPay>	DownPayToWineSupplier
	<eCaterer, WineSupplier, FinalPay>	FinalPayToWineSupplier
<FoodSupplier, eCaterer, Food>		FoodToCateringCompany
<eCaterer, FoodSupplier, Money>		FinalPayToFoodSupplier

8. How do you order the Economic Events and the Atomic Economic Events? (Add only < symbols in the matrix)

	MealToCustomer	DownPayFromCustomer	FinalPayFromCustomer	WineToCateringCompany	DownPayToWineSupplier	FinalPayToWineSupplier	FoodToCateringCompany	FinalPayToFoodSupplier
MealToCustomer			V					
DownPayFromCustomer	V		V					
FinalPayFromCustomer								
WineToCateringCompany	V					V		
DownPayToWineSupplier				V		V		
FinalPayToWineSupplier								
FoodToCateringCompany	V							V
FinalPayToFoodSupplier								

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

Fig. 12~~12~~. Questions and answers for the example in phase 2

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 business order can be completed with reservation and booking as follows.

9a. Do you reserve economic resource, ER_1 before reserving economic resource, ER_2?
9b. Do you book economic resource, ER_1 before booking economic resource, ER_2?
9c. Do you book economic resource ER_1 before reserving economic resource ER_2?

An example answer to the questions is given in [**Fig. 13~~Fig. 13~~**]. The resulting partial order from phase 2 is the following (< means precedes):

{ MealToCustomer < FinalPayFromCustomer,
DownPayFromCustomer < MealToCustomer,
DownPayFromcustomer < FinalPayFromCustomer,
WineToCateringCompany < MealToCustomer,
WineToCateringCompany < FinalPayToWineSupplier,
DownPayToWineSupplier < WineToCateringCompany,
DownPayToWineSupplier < FinalPayToWineSupplier,
FoodToCateringCompany < MealToCustomer,
FoodToCateringCompany < FinalPayToFoodSupplier,
Book-Wine < Book-Food,
Reserve-Wine < Reserve-Food}

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

9. For each atomic economic event, identify the ones needed with counter offer handling.
.Wine supply with counter offer handling
.Food supply with counter offer handling

For each economic resource with counter offer handling, fill following matrix.

Column → / Row ↓	Wine	Food
Wine	-	9a. Book-Wine < Book-Food? Yes 9b. Reserve-Wine < Reserve-Food? Yes 9c. Book-Wine < Reserve-Food? No
Food		-

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

Fig. 13~~13~~. Questions and Answers for Reservation and Booking

5.3 Phase 3 - Process Order

In phase 3, the designer will extend the business 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 (modeled by an ERRD or EROD diagram in phase 4). 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 that B has done its corresponding economic event (or some preparation). Note that this check will be done only if the business 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 business order, ask:

10a. Do you require that In be performed before you commit to perform Out?
10b. 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 business order, ask:

11a. Do you require that Means be performed before you commit to perform Out?
11b. Do you require a commitment for Means before you commit to perform Out?
11c. Do you require that In be performed before you request Means?

An example of answers to these questions is given in **Fig. 14**~~**Fig. 14**~~. All answers will result in an extension to the business 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:

$$PO = BO \cup \{ decl(DownPayfromCustomer) < com(MealToCustomer),\ com(WineToCateringCompany) < com(FinalPayToWineSupplier),\ com(FoodToCateringCompany) < com(FinalPayToFoodSupplier),\ com(WineToCateringCompany) < com(MealToCustomer),\ decl(DownPayFromCustomer) < dir(WineToCateringCompany),\ com(FoodToCateringCompany) < com(MealToCustomer),\ decl(DownPayfromCustomer) < dir(FoodToCateringCompany)\}$$

BO is the business order derived in phase 2 for the example.

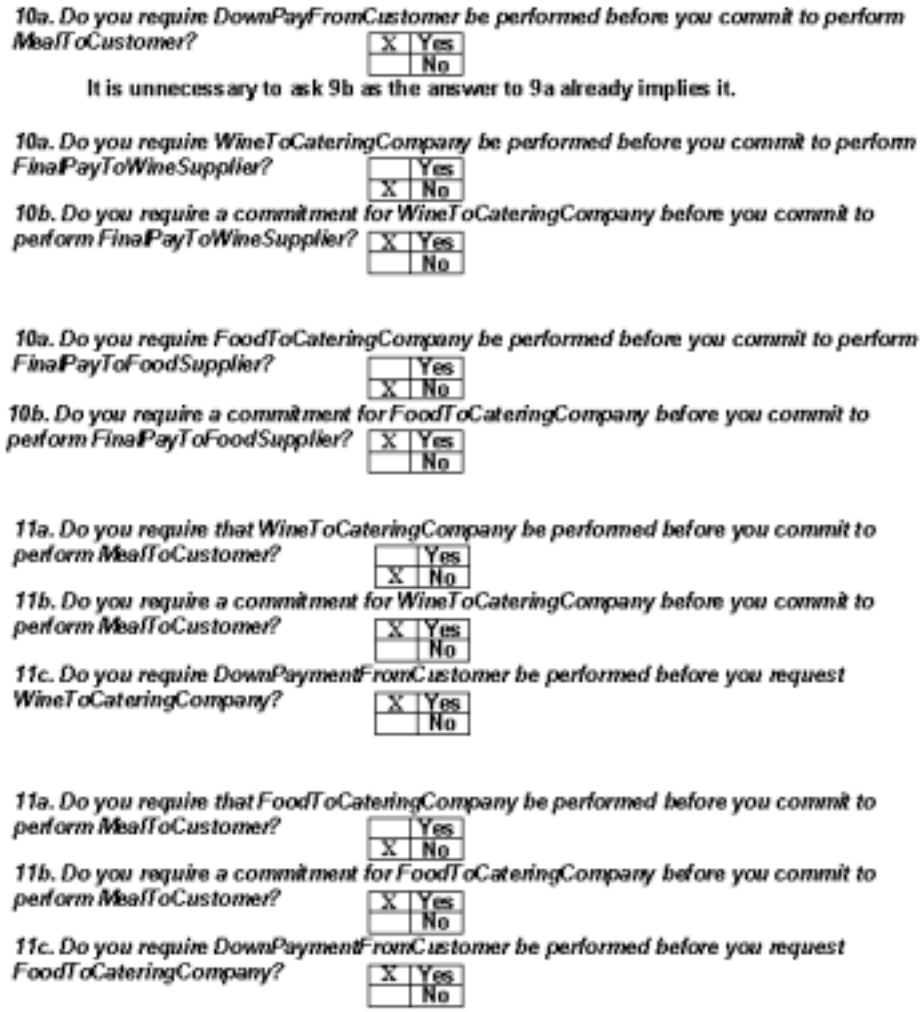


Fig. 14~~14~~. Example of answers to question sets 9) and 10)

5.4 Phase 4 - Mapping Process Order to BML Process Model

After completing phase 3, the designer ends up with a set of ordered communication acts. For each inequality in this partial order, we have defined a rule that completes inter-diagram communication by connecting IN-ports of one diagram with OUT-ports of another diagram.

The complete set of rules to map a Process Order to a BML Process Model is not listed here due to space limitations but it can be found in [17]. For illustrative purposes, rules necessary to handle inter-dependencies between reservation and booking of economic resources are listed below.

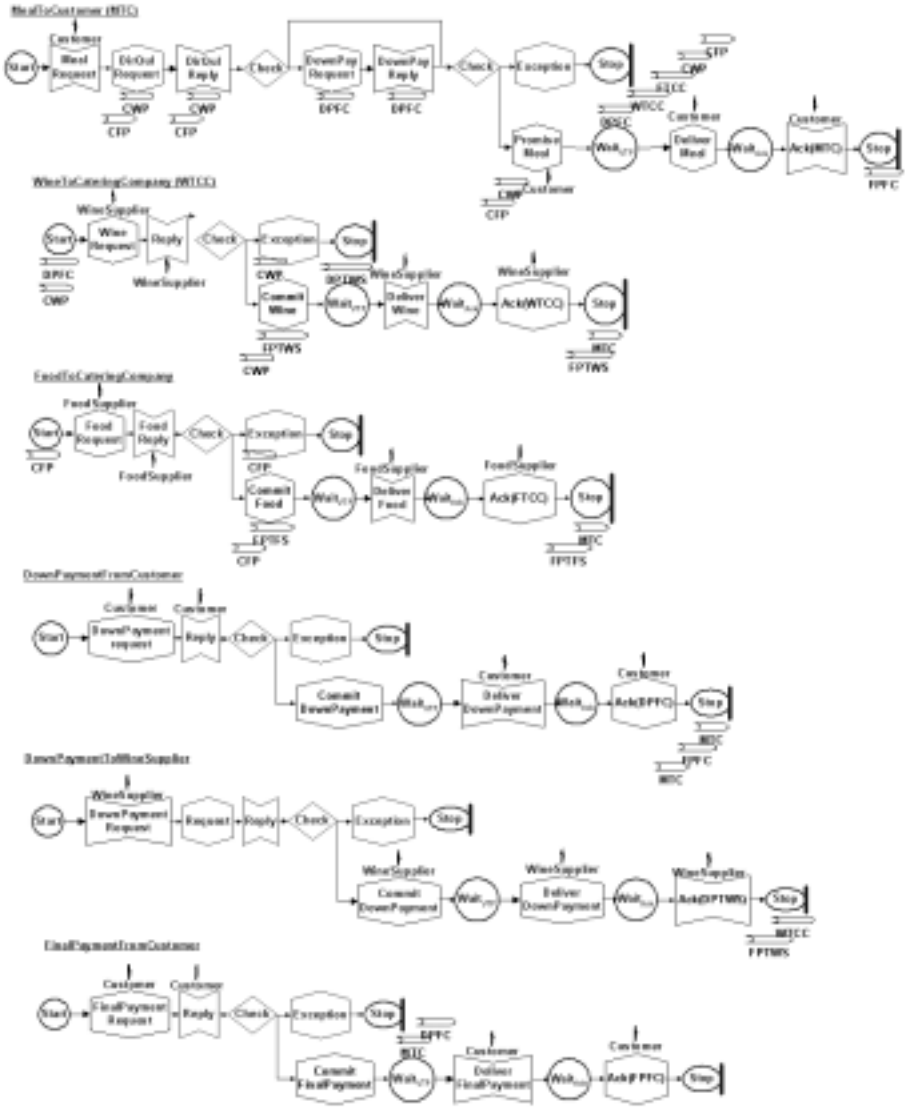
$\forall$ $Book\text{-}ER_1 < Book\text{-}ER_2$
add($Diagram_2$, {Stop, $Diagram_1$})
add($Diagram_1$, {$Wait_{BR}$, $Diagram_2$})

$\forall$ $Reserve\text{-}ER_1 < Reserve\text{-}ER_2$
add($Diagram_2$, {Confirmation, $Diagram_1$})
add($Diagram_1$, {$Wait_{RR}$, $Diagram_2$})

$\forall$ $Book\text{-}ER_1 < Reserve\text{-}ER_2$
add($Diagram_2$, {Stop, $Diagram_1$})
add($Diagram_1$, {$Wait_{RR}$, $Diagram_2$})

The operation add(A, {B, C}) means place diagram A at B position of diagram C. Similarly there is another operation remove(A, {B, C}) which removes existing diagram A at position B of diagram C in order to remove unnecessary redundant communications.

The initial BML model can be generated by adding relevant ERRD or EROD for each identified economic event and also pairs of PCD and COD for necessary economic resources (Ers). When instantiating EROD's relevant qualification path has to be selected. Then by applying the rules the final BML model can be generated. The result after this generation for our example is given in **Fig. 15**~~**Fig. 15**~~.



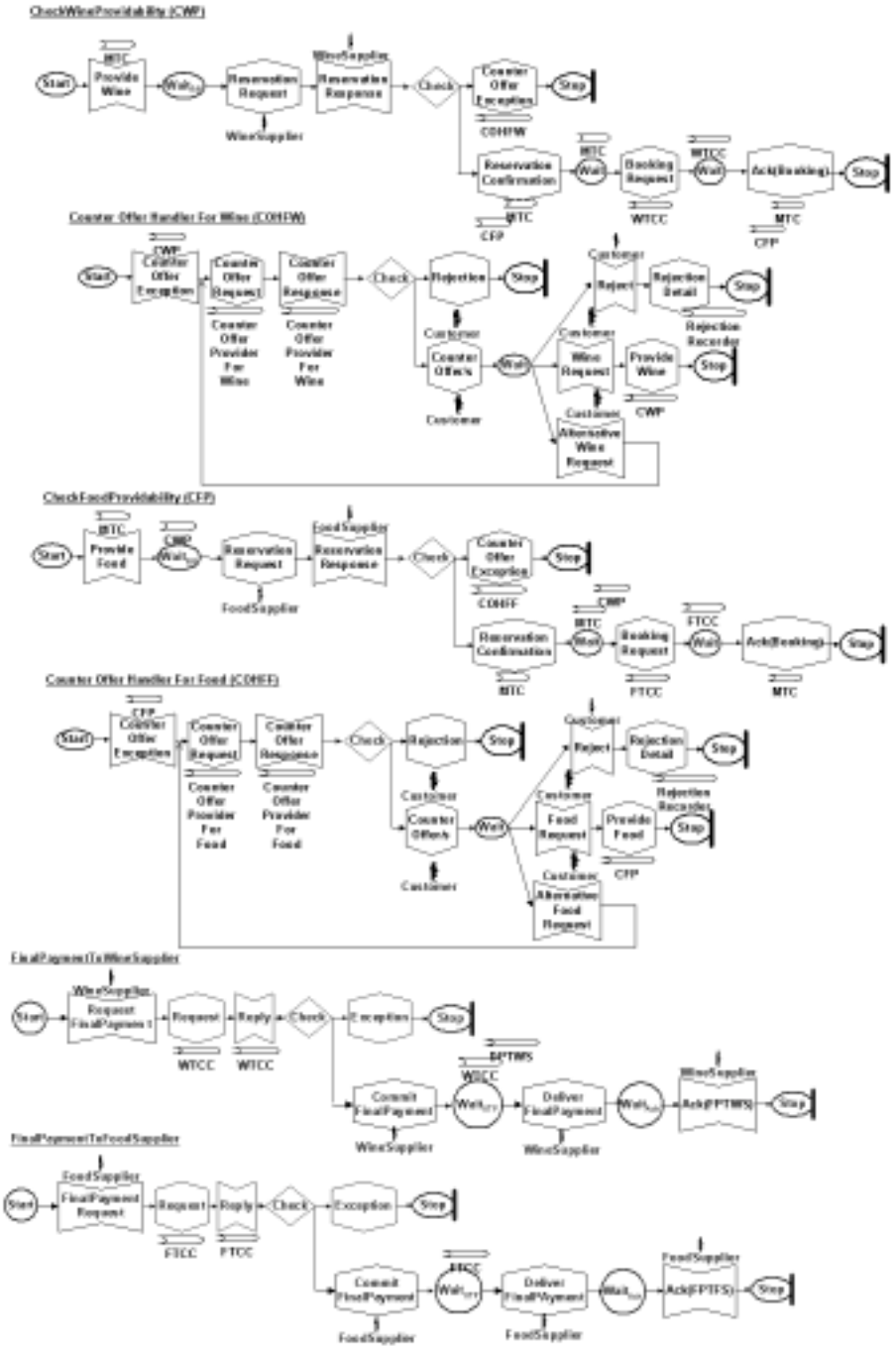


Fig. 1515. Generated Final BML process model with completed inter diagram communications.

The entire inter-process and partner communication for the example discussed here can be shown in the static diagram shown in [Fig. 16~~Fig. 16~~] .

Fig. 16~~16~~. BML static diagram for the eCatering example

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.

References

1. Austin J.L., *How to do things with words*, Oxford, 1962.
2. Belina F., Hogrefe D. and Amarddeo S.: "*SDL with Applications from Protocol Specification*", Carl Hanser Verlag and Printice Hall International UK 1991.
3. Goldkuhl, G.: "Generic Business Frameworks and Action Modeling", *First International Workshop on Communications Modeling - The Language/Action Perspective*, Springer Verlag 1996.
4. Gordijn J., Akkermans J. M. & Vliet J. C.: "Business Modeling is not Process Modeling", *eCOM2000 workshop, 19th International Conference on Conceptual Modeling 2000*.
5. Gordijn J., Akkermans J. M. & Vliet J. C.: "What's in an Electronic Business Model? ", *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference*, Springer-Verlag 2000.
6. Information Systems, Volume 26, Issue 3 SummaryPlus, May 2001 Article Pages 165-184.
7. Jayaweera P., Johannesson P., & Wohed P.: "From Business Model to Process Patterns in e-commerce", The Sixth International Workshop on the Language-Action Perspective on Communication Modelling 2001, Montreal, Canada.
8. Johannesson P.: "A Language/action based Approach to Information Modeling", in *Information Modeling in the New Millennium*, eds. M. Rossi and K. Siau, IDEA Publishing, 2001.
9. 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.
10. Raul Medina et al.: "The Action Workflow Approach to Workflow Management Technology", Proceedings of 4th Conference on Computer Supported Cooperative Work, ACM Press, 1992.
11. Malone et al.: "Towards a handbook of organizational processes", MIT eBusiness Process Handbook http://ccs.mit.edu/21c/mgtsci/index.htm.
12. Porter M. E.: *"Competitive Advantages. Creating and Sustaining Superior Performance"* The Free Press 1998.
13. SDL Standards, http://www.sdl-forum.org/Publications/Standards.htm.

14. Searle J.: *"Speech Acts - An Essay in the Philosophy of Language"*, Cambridge University Press 1969.

15. Taylor, J. R., Groleau, C., Heaton, L. and VAN EVERY E. J.: *"The Computerization of Work : A Communication Perspective"*, Thousand Oaks CA: Sage.

16. Taylor J.: "The Limits of Rationality in Communication Modeling – a Semiotic Reinterpretation of the Concept of "Speech Act"", *Third International Workshop, The Language Action Perspective on Communication Modeling*, eds. G. Goldkuhl et.al. 1998.

17. Technical Report, "Mapping Function to Generate BML Process Model", url: http://www.dsv.su.se/~prasad/html/MapFun.doc

18. UN/CEFACT, url: http://www.unece.org/cefact/ece_top.htm, 10th of May 2001.

19. Viewlocity, url: http://www.viewlocity.com/solutions/frame_index.html.

20. Weigand H., van den Heuvel W. and Dignum F.: "Modeling Electronic Commerce Transactions – A Layered Approach", *Third International Workshop, The Language Action Perspective on Communication Modeling*, eds. G. Goldkuhl et.al. 1998.

21. Weigand H., Moor A. de & Heuvel W-J. van den, "Supporting the evolution of workflow patterns for virtual communities", in *Proceedings of the 33rd Hawaii International Conference of System Sciences*, 2000.

22. Winograd, T. & Flores, F., *"Understanding Computers and Cognition: A New Foundation for Design"*, Ablex, Norwood, N.J 1986.

23. Wohed, P., "Conceptual Patterns for Reuse in Information Systems Analysis", B. Wangler and L. Bergman.(Eds.), *Proc. from the 12th International Conference on Advanced Information Systems - CAiSE 2000*, LNCS 1789, Springer, pp. 157-175, 2000

24. Wohed, P., "Tool Support for Reuse of Analysis Patterns – a Case Study", A. Laender, S. Liddle and V. Storey (Eds.), *Proc from the 19th Int. Conference on Conceptual Modeling - ER2000*, LNCS 1920, Springer, pp 196-209, 2000.