

# Using Business Transactions to Analyse the Consistency of Business Process Models

Artur Caetano  
IST, Technical University of  
Lisbon and INESC-ID.  
Rua Alves Redol 9, 1000-029  
Lisboa, Portugal.  
[artur.caetano@ist.utl.pt](mailto:artur.caetano@ist.utl.pt)

Aurélio Assis  
IST, Technical University of  
Lisbon and INESC-ID.  
Rua Alves Redol 9, 1000-029  
Lisboa, Portugal.  
[aurelio.assis@ist.utl.pt](mailto:aurelio.assis@ist.utl.pt)

José Tribolet  
IST, Technical University of  
Lisbon and INESC-ID.  
Rua Alves Redol 9, 1000-029  
Lisboa, Portugal.  
[jose.tribolet@ist.utl.pt](mailto:jose.tribolet@ist.utl.pt)

## Abstract

*This paper presents a method to analyse the consistency of a process model according to business transactions. Business transactions focus on specifying the pattern-based behaviour resulting from the collaboration between actors and systems while services and products are being requested and provided. The method makes possible assessing the consistency of a business process in terms of the business transactions that can be inferred from it. To do so, it takes as input a process model that is converted to a transactional model. The transactional model is then analysed and revised so that all transactions become consistent according to the patterns of DEMO. Finally, the original process model is revised to comply with the transactional model. As a result, the revised business process becomes consistent with the corresponding transactional model.*

## 1 Introduction

Business process modelling is presently supported by several standards, notations, languages and tools. The resulting representations are used to support tasks such as the analysis and communication of inter- and intra-organizational business processes as well as the development of organizational support systems [1-3]. The project reported in this paper is motivated by the need to produce consistent and complete business process models as a means to reduce the ambiguity of using such models to support the aforementioned tasks. This need arises from the fact that process models that focus on specifying the flow of activities and information, such as BPMN [4], do not provide the means to assess the actual consistency and completeness of a business process. This is partially due to the lack of formal semantics behind these languages and to the ambiguous and unclear descriptions of their constructs [5, 6]. These problems become evident when rigorous and detailed models are required to guide the specification and development of the services and information system that support a business process. An example of such ambiguity is the usage of natural language as the only means to convey the meaning of the activities and informational objects that

pertain to a process. Besides, business process modelling methods tend to focus on aligning business and technological concepts at a high level of detail but do not prescribe the principles to design complete business process [5]. The combination of these factors creates the need to define techniques that facilitate the consistent design of new business processes or provide the means to analyse process models with the goal of continuously improving their consistency while responding to the stakeholder's needs.

This paper proposes using the concepts set forth by the Design & Engineering Methodology for Organizations (DEMO) [7] as a foundation to analyse the consistency and completeness of business processes. We consider a business process model to be *consistent* if its activities can be translated to business transaction steps. This entails making the responsibilities behind each activity clear as well as the communication patterns between the actors of the process. But a business process model can be consistent but still be missing activities that are required to fully produce the intended transactional results. This means that a *complete* process must specify all required activities. Thus, a process model that is consistent and complete explicitly and fully specifies an end-to-end collaboration pattern between a service requester and a service provider.

DEMO is a methodology for the design, engineering, and implementation of organizations that focus on explaining how and why people cooperate and communicate. It considers that entering into and complying with commitments is the key principle that operates an organization. Some authors state that DEMO defines a robust enterprise engineering approach as it provides a formal yet simple conceptualization of an organization that can be used a point of departure for its implementation [1, 8]. DEMO can also deliver models which can be formally assessed and executed on DEMO automata [9]. However, DEMO ends where BPMN and other transformational business process modelling languages start. Thus, this paper proposes using DEMO to help constructing consistent and complete BPMN process models. Yet, the approach can be adapted to other modelling

languages that focus on specifying the control and data flow of a process.

Nuffel et al. [1] state that the design of an enterprise may be expressed with an ontology containing social interaction concepts which in its implementation includes an ontology for information systems and an ontology for documental systems. It can also be argued that documental and informational ontologies are subsets of enterprise ontology, thus leading to the conclusion that instead of investigating a one-to-one mapping between concepts, a many-to-many mapping has also to be researched. These authors thus argue that DEMO can provide a formal foundation to specify BPMN models. Using enterprise ontology as underlying domain ontology results in explicitly specified BPMN models, mainly because enterprise ontology delivers principles and constraints to which BPMN should adhere. Moreover, DEMO can also be used to assess existing BPMN models with the purpose of verifying their completeness and consistency. The main contribution is thus combining the representational capability of the BPMN standard with the formal correctness of the DEMO principles.

The next section of the paper describes the DEMO models and the underlying  $\Psi$ -theory. Section 3 describes the proposed iterative method for business process analysis and revision. Section 4 exemplifies the method by applying its steps to a BPMN process model. Finally, section 5 draws some concluding remarks.

## 2 DEMO and the $\Psi$ -theory

The Design & Engineering Methodology for Organizations (DEMO<sup>1</sup>) is a methodology for the design, engineering, and implementation of organizations. Its models are considered comprehensive and coherent as they operate as an integrated whole [7]. DEMO is formally grounded on the PSI- or  $\Psi$ -theory<sup>2</sup> that combines concepts from the language/action perspective [10] and speech act theory [11]. The  $\Psi$ -theory explains how and why people cooperate and communicate. It postulates that the operation of an organization can be expressed by a specification of the commitments that the organizational subjects (i.e. the organizational actors) enter into and comply with. DEMO draws from this theory to specify an enterprise ontology that captures the understanding of an enterprise's construction and operation in a way that is fully independent of its realization and implementation. The enterprise ontology corresponds to the highest-level constructional model of an enterprise, with the implementation model being the lowest one.

<sup>1</sup> <http://www.demo.nl/>

<sup>2</sup> PSI stands for Performance in Social Interaction.

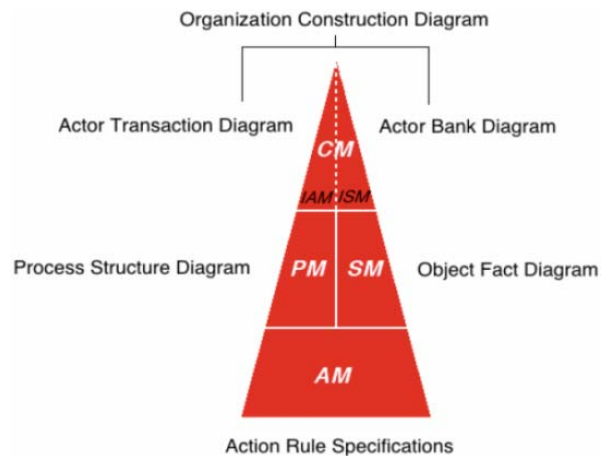


Figure 1. The ontological triangle (adapted from [7]).

The  $\Psi$ -theory underlines the notion of enterprise ontology [7]. An enterprise is here defined as a heterogeneous system in the category of social systems. Being a social system means that its elements are social individuals. The  $\Psi$ -theory consists of four axioms that are briefly described next since they are significant to understand the DEMO models as well as the method put forward in this paper. The DEMO artefacts are expressed a set of interrelated models that, as a whole, specify the ontology of the enterprise. These models (Construction Model CM, Process Model PM, State Model SM and Action Model AM) are arranged as an “ontological triangle” illustrated in Figure 1 and will be described at the end of this section.

### 2.1 The $\Psi$ -theory Axioms

The  $\Psi$ -theory comprises four core axioms that specify how individuals communicate and establish commitments. The next sections introduce the operation, transaction, composition and distinction axioms.

**2.1.1. Operation Axiom.** The operation axiom states that the operation of an enterprise is constituted by the activities of actor roles. An actor role is an element of authority and responsibility which is fulfilled by subjects. The subjects perform two kinds of acts: *production acts* (P-acts) and *coordination acts* (C-acts). By performing P-acts, the subjects contribute to bringing about the function of the organization, i.e. the goods and services that are delivered by the enterprise. The realization of a production act is inherently material or immaterial. Examples of material acts are manufacturing acts, as well as storage and transportation acts. By performing C-acts, the subjects enter into and comply with commitments regarding P-acts. So C-acts are the way in which cooperation between subjects is accom-

plished and made explicit. An (elementary) actor role is defined as the authority to perform one particular kind of P-act. A subject in its fulfilling of an actor role is called an actor.

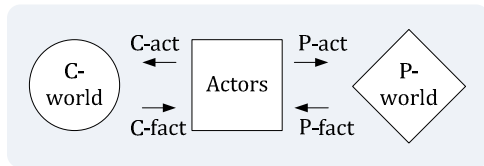


Figure 2. Notation for actor, coordination and production (adapted from [7]).

P-acts and C-acts produce results. The result of a production is a production fact (P-fact) and the result of coordination act is a coordination fact (C-fact). For instance, the action of packing an order (a material P-act) results in a pack with the order items (the corresponding P-fact). The production and coordination worlds contain all the corresponding acts and facts. The notation to represent these concepts is summarized in Figure 2.

**2.1.2. The Transaction Axiom.** The transaction axiom states that C-acts and P-acts always occur in a particular pattern. The standard pattern consists of two fundamental conversations: an *order* conversation (actagenic) which is followed by a *result* conversation (factagenic). A conversation is defined as a sequence of coordination acts between two actor roles that are aimed at achieving a well-defined result. So, a transaction evolves in three phases: the order phase (O-phase) where the request is made, the actual execution phase (E-phase) that produces the requested result, and the final result phase (R-phase) where the initiator is notified of the result. The actor role that starts the order conversation is the *initiator*. The role that executes the request, produces the fact and responds to the initiator is the *executor*.

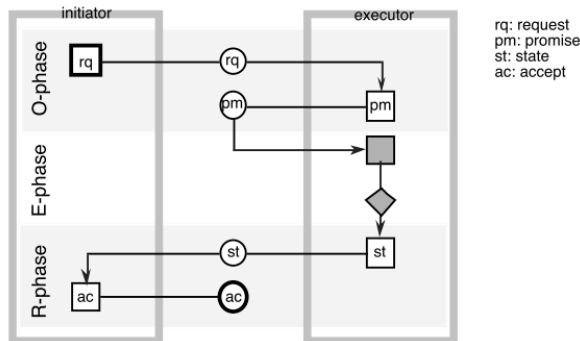


Figure 3. The basic transaction pattern (adapted from [7]).

The basic transaction pattern is depicted Figure 3. The initiator starts by initiating the conversation with the executor. This is accomplished by making a request (rq) to the executor to produce a specific fact (P-fact). This

P-fact is the result of the transaction. The executor then promises (pm) the initiator that he agrees with the request and that he is committed to produce the result. Thus, the order phase defines the initial contract or commitment between the initiation and executor. After the promise, the executor produces the P-fact. This production action is depicted by the two dark symbols: the dark square depicts a P-act and the dark diamond a P-fact. The remaining white symbols represent coordination: the white squares are C-acts whereas the white circles are C-facts. Note that the white and grey squares actually represent actors that can be either performing coordination or production acts (cf. Figure 2). The dark colour also indicates that the production is a black-box specification as it is not relevant how the executor actually produces the results. This is because DEMO focuses on the conversations between actors and their commitments and not how such commitments are operated. The next step is the executor communicating to the initiator that the result is complete (state). Finally, the initiator accepts (ac) the result, successfully ending the transaction.

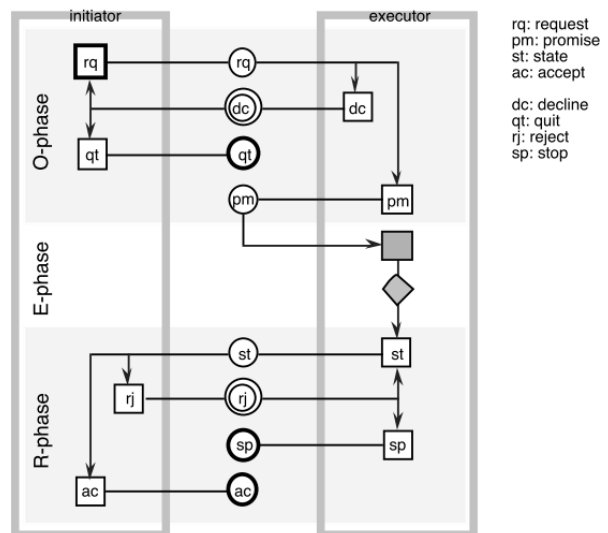


Figure 4. The standard transaction pattern (adapted from [7]).

The basic transaction pattern assumes that all conversations have a successful outcome (e.g. the executor always promises to fulfil the initiator's request, the initiator always accepts the results, and the executor never aborts the production of the result after promising to produce it). However, DEMO defines a complete transaction pattern that extends the basic pattern with four additional cancellation patterns (v. Figure 4). According to Dietz [7], the standard transaction pattern is considered universal and complete as it covers all possible communicative actions, including possible rollback situations that may occur during the order, execution and result phases.

**2.1.3. The Composition Axiom.** The composition axiom provides the basis for formally defining a business process. It states that a business process is a collection of causally related transactions, so that the starting step is either a request performed by an actor role outside the organizational environment (an external activation) or it is a request by an internal actor role to itself (a self-activation). The result of a successful transaction is always the creation of a P-fact by the executor. Conversely, this also means that every original new fact in the P-world is brought about as the result of a successful transaction. Thus, the composition axiom states how all the production facts occurring within an organization and the corresponding transactions are interrelated.

**2.1.4. The Distinction Axiom.** The fourth axiom of the  $\Psi$ -theory states that there are three distinct human abilities playing a role in the operation of actors, called *performa*, *informa*, and *forma*. These abilities regard communicating, creating things, reasoning and processing information.

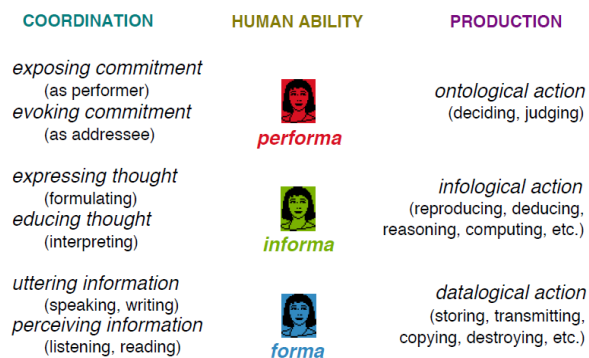


Figure 5. The distinction axiom (adapted from [7]).

## 2.2 The DEMO Models

DEMO comprises four fundamental models: Construction Model, Process Model, State Model and the Action Model. These will be described next.

**2.2.1. The Construction Model.** The construction model (CM) of an organization specifies its composition, its environment, and its structure. The composition and the environment are both a set of actor roles fulfilled by humans individually or collectively. The interaction structure of an organization therefore consists of the transaction types in which the identified actor roles participate as initiator or executor. It is expressed as an Actor Transaction Diagram (ATD) and a Transaction Result Table.

**2.2.2. The Process Model.** The process model (PM) specifies the state space and the transition space of the C-world and thus the set of lawful or allowed sequences of states in the coordination world. This is to a large extent determined by the transaction pattern. Since every transaction in the C-world consists of the creation of a C-fact and since there is a one-to-one relationship between this C-fact and the causing C-act, these C-acts are also contained in the PM. A C-fact and its causing C-act are collectively called a *process step*. The PM is partially based on the information defined on the CM concerning which actor roles perform the C-acts. The coordination acts performed by each role are designated the responsibility areas of that actor role.

Figure 6 exemplifies a process that is composed of two transactions (T01 and T02). T01 is a transaction executed by actor A01 that is externally activated by actor CA01 who requests it. To complete T01 an additional enclosed transaction T02 must be executed by actor CA02. So, there is a total of three roles involved in these two transactions. The transaction T01 is initiated by CA01's request. A01 either declines or promises this request. In case of a promise (T01/pm), A01 initiates a second transaction T02 with a new request to CA02. A01 could also initiate the production of the result T01.

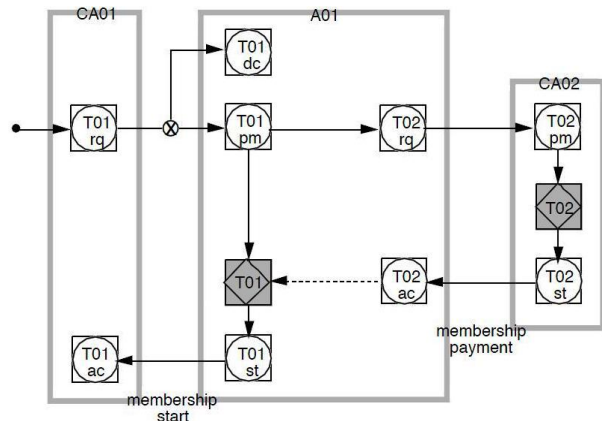


Figure 6. A process structure diagram (adapted from [7]).

However, executing this P-act depends on the conclusion of the acceptance of T02. This is depicted by the dotted dependency arrow between T02/ac and T01. Thus, the result T01 is pending on the conclusion of T02/ac. As a consequence of T02/rq, CA02 promises (T02/pm) to A01 and then produces the results of T02. After completion, it states the fact to A01 (T02/st). A01 accepts CA02's results (T02/ac). A01 can now start producing the results of T01 that were waiting for the conclusion T02/ac. Finally, A01 states to CA01 that the results are ready (T01/st). The transaction ends when the initiator CA01 accepts (T01/ac) the result of T01.

**2.2.3. The State Model.** The state model (SM) is the specification of the state space of the P-world. It consists of specifying the object classes, the fact types, and the result types, as well as the existential laws that hold. An SM is expressed with an Object-Fact Diagram (OFD) using Object Role Modelling [12]. The SM also includes an Object Property List that is able to specify fact types that are mathematical functions.

**2.2.4. The Action Model.** The action model (AM) is the most comprehensive aspect model and integrates the concepts defined on the other aspect models (PM, SM, and CM). Therefore, the other models can be derived from the AM. This model describes the rules that apply to every transaction.

```

when membership start of [membership] is promised
      with current year is [year]
if member of [membership] applies for reduced fee for [year]
then reduced fee approval of [membership, year]
      must be requested
else membership fee payment of [membership, year]
      must be requested with standard fee of [year]
  
```

Figure 7. A DEMO action model (adapted from [7]).

### 3 The Method

This section proposes an iterative method for the analysis of business process models that aims to improve their consistency and completeness. We consider a business process model to be consistent if its actions do not contradict the transactional axiom (cf. section 2). A business process model is deemed complete if it contains all the actions that are required to produce the intended transactional result. So a process model that is both consistent and complete specifies all the actions that are required to produce a specific result and does so according to the transactional axiom. Note that completeness and consistency apply to any level of detail of the process. Therefore, if a business process is functionally decomposed then all levels of detail must be consistent with the transactional model.

This method attempts to address one problem pertaining to process modelling: the inability to assess whether a process model fully specifies the conversation steps between the actors involved in it. If a process model fails to do so, then the specification of how actors enter into and comply with commitments may be ambiguous or not be expressed at all. This means that the inter- or intra-organizational areas or responsibility may be unclear. An example of this is when an actor requests the production of some result to another actor but the actual contract that regulates the transaction is either implicit or incompletely specified. This means that the transactional steps such as promising, stating or accepting a result may be left unspecified in the process model.

As a result, this hinders the procedure of checking and tracing who is responsible for doing what and even why some result is actually being produced.

The method is based on DEMO and comprises two main phases. It also assumes that the input business process model adheres to the transformational paradigm [13]. Throughout this paper we will actually assume the input model to be a BPMN process diagram. A transformational process specification shows how resources are handled by the process activities in order to obtain some result. The sequence of activities is specified using control and data flow mechanisms. Languages such as BPMN specify a process as a sequence of named activities whose flow follows decision points (gateways) and that are performed by specific actors.

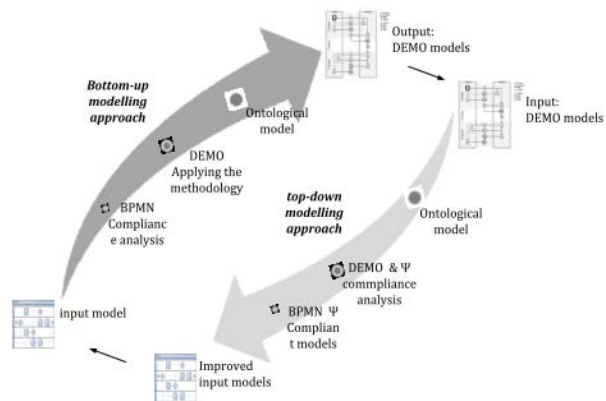


Figure 8. The bottom-up and top-down iterative analysis.

The bottom-up phase starts with an existing business process model, such as a BPMN process diagram, and constructs a DEMO process model that is represented as one or more process structure diagrams (PSD). The goal of the bottom-up phase is not to translate the original model to a DEMO process model but to identify its production and coordination acts. These abstracted production and coordination acts are then modelled in DEMO and analysed according to the  $\Psi$ -theory axioms to assess their consistency and completeness. The bottom-up phase is therefore the first step to capture the coordination and production acts of a process model and enables to check whether a process model complies with the DEMO transaction pattern.

The top-down phase is a gap analysis phase where missing or superfluous actions are identified and the corrected DEMO model complies with the  $\Psi$ -theory. This corrected DEMO model can then be used as a tool to review the original process model so that its embedded coordination and production actions comply with the DEMO model. The bottom-up and top-down phases are applied iteratively to the original process model. This enables the model to be incrementally analysed and refined as illustrated in Figure 8. The remainder of this

section elaborates on these two phases. An example of application of the method is described afterwards.

### 3.1 The Bottom-Up Phase

The bottom-up phase starts with a BPMN process model and produces a DEMO process model that abstracts the coordination and production acts depicted on the input model.

**3.1.1. Step 1: Analyse the Input Process Model.** The goal of the first step is to analyse the design artefacts used to represent the business process model. It analyses the activities and classifies them according to the operation axiom and distinction axiom (cf. section 2). As a result each activity is classified as a production or coordination act (operation axiom) and also as a *performa*, *informa*, or *forma* speech act (distinction axiom). Furthermore, the operation axiom also discriminates the actor roles involved in the process.

Consider the following examples: an activity that entails sending an electronic message to someone is classified as a coordination act since it involves the communication between actors. It is also classified as a *forma* act because it represents a source uttering a message to a recipient. An activity that archives that message is also classified as *forma* but it is a production act because it is generating a new production fact (the archived message). And an activity that counts the number of messages archived on a given date is a production act as it generates a new production fact (the message count) but it is classified as an *informa* speech act since it is computing a result.

The result of this step is a traceable list that maps the coordination and production acts and actors to the original process model from where they were sourced.

**3.1.2. Step 2: Generate the DEMO Models.** The previous step analyses the input process model according to the operation axiom and distinction axiom. This enables dividing the input process activities in two groups: *performa* acts and *forma* plus *informa* acts. This step starts by focusing on the *performa* coordination and production acts since this level represents the ontological actions. The goal is now to further classify each *performa* coordination act according to the transaction axiom. This means each business process activity will be classified as a *request*, *promise*, *state* or *accept* coordination act according to the basic transaction pattern. If the process is analysed according to the standard transaction pattern then the *decline*, *reject*, *stop* or *quit* acts are also considered.

For instance, an activity “Place Order” is as a *request* coordination act as it is performed by the initiator to start

a new transaction. The activity “Receive Ordered Product” is an *accept* coordination act as it indicates the initiator acknowledged and accepted the result of the transaction.

The classification of the coordination acts along with the control and data flow restrictions that are specified on the input process model are used to generate the corresponding DEMO process model.

**3.1.3. Step 3: Revise the DEMO Models.** The DEMO model reviewing is about adding acts found to be missing during the previous step. These missing acts are the ontological acts that are not described explicitly in the analysed input business process. The model is revised according to the transaction axiom so that every transaction goes through the steps defined in the O-phase, E-phase, and R-phase. Next, the application of the composition axiom ensures that all of the transaction steps follow a logical sequence according to the pattern.

This is the last step on the bottom-up phase and creates a set of revised DEMO diagrams, namely a process structure diagram (process model) and an actor-transaction diagram (construction model).

### 3.2 The Top-Down Phase

In this phase, the DEMO models produced as the output of the previous phase are used as a means to check the compliance of the original BPMN process models with the revised DEMO models (step 1). The results of this assessment are then used to revise the input model so that it becomes complete and consistent with the DEMO model (step 2).

**3.2.1. Step 1: Assess the compliance.** This step performs a gap analysis that identifies the ontological acts in the revised DEMO model that are not reflected in the input business process model.

**3.2.2. Step 2: Revise the Input Process Model.** The original BPMN process model is revised so that its activities match the missing ontological acts. The revised process becomes  $\Psi$ -theory compliant, meaning it is now consistent with the transactional model and complete as it explicitly contains all transactional steps.

## 4 Example

Figure 9 depicts a business process represented in BPMN. In this process, a buyer has the goal of buying a product from a seller. To do so, it places an order to a seller who checks its validity. The seller then sends an invoice to the buyer. After receiving the payment the

seller sends the buyer the ordered product. This example process is intentionally being specified at a high level of functional detail and it excludes error conditions and compensation actions to simplify the application of the method. The application of the method to a complex process is similar but as expected entails a lengthier analysis. Nevertheless, the DEMO ontological models have proven to offer a significant reduction of complexity compared to the corresponding implementation models when applied to large real business processes [14-16]. The next sections show the application of each step of the method to this example.

#### 4.1 Application of the Bottom-Up Phase

**4.1.1. Step 1.** The application of the first step of the bottom-up phase classifies one of the activities (Check Order) as infological (*informa*) according to the distinction axiom. The remaining activities are classified as ontological (*performa*) acts. As described in the previous section, the coordination acts are further classified according to the transaction axiom (e.g. Place Order is classified as T01/request and Receive Invoice as T02/promise). The resulting labelled activities are shown in Figure 9.

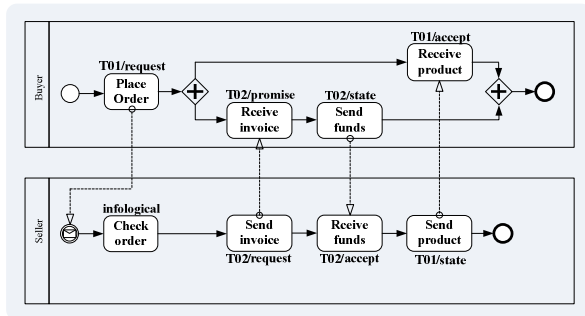


Figure 9. Labelled input model (step 1).

**4.1.2. Step 2.** The second step of the bottom-up phase derives a DEMO process from the labelled business process model. The resulting model is shown in Figure 10. It can be observed that this model is incomplete as several acts are missing.

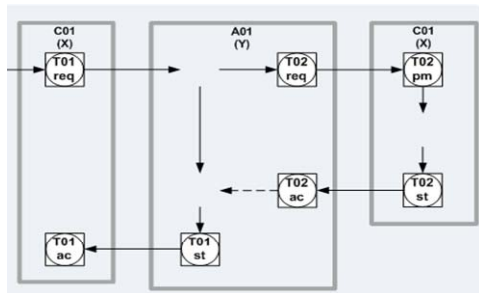


Figure 10. Incomplete process structure diagram derived from the input model (step 2).

**4.1.3. Step 3.** This step revises the process model according to the transaction and composition axioms. The result is the process model shown in Figure 11 which adds the acts missing from Figure 10 (T01/pm, T01, T02) according to the transaction pattern. This revised model is now complete and consistent with the  $\Psi$ -theory axioms.

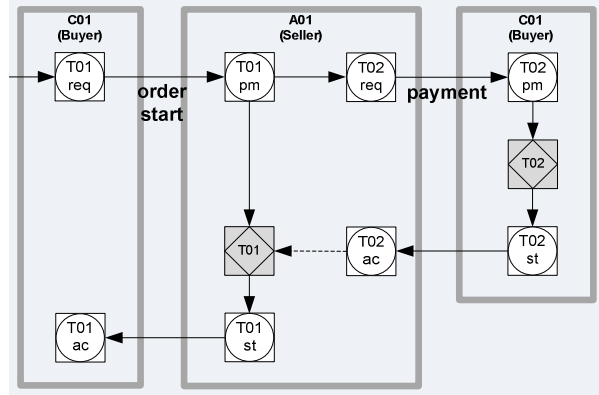


Figure 11. Revised process structure diagram (step 3).

The actor-transaction diagram (ATD) depicted in Figure 12 shows the actors responsible for initiating and executing a transaction. The buyer is responsible for initiating the transaction T01 (Order) which is executed by the seller (the executor role is depicted by a black diamond on the association end). Conversely, the seller initiates transaction T02 (Payment) which is executed by the buyer.

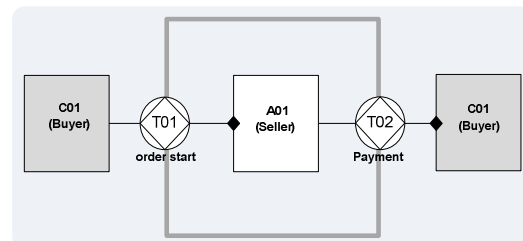


Figure 12. Construction model (step 3).

#### 4.2 Application of the Top-Down Phase

This section describes the two major steps of the top-down phase which use the DEMO models produced in the previous phase to analyse and revise the original BPMN process model according to the  $\Psi$ -theory axioms.

**4.2.1. Step 1.** This step identifies what acts are represented in the revised DEMO model but are missing from the original BPMN process. Figure 13 emphasizes the three missing acts: T01/pm and the production of the T01 and T02 results.

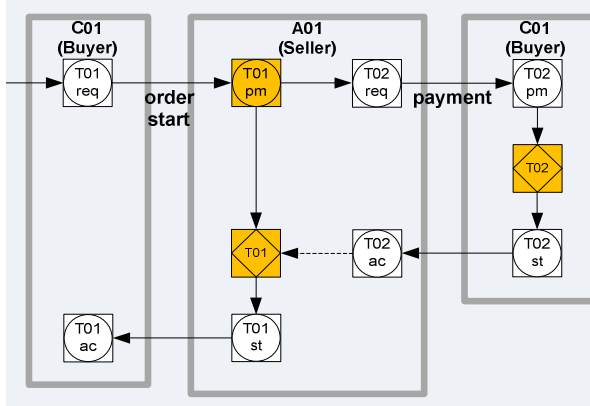


Figure 13. Gap analysis: missing steps in the DEMO process model (step 1).

**4.2.2. Step 2.** This step revises the input process model according to gaps identified in the previous step (T01/pm, T01, T02). The revised labelled BPMN process is depicted in Figure 14. The changes imply adding three new BPMN activities, one for each missing ontological act: T01/promise, T01/execute and T02/execute. Note that the name of these activities is actually irrelevant at this stage. The revised BPMN process is now consistent and complete with the  $\Psi$ -theory axioms since all essential speech acts are explicit.

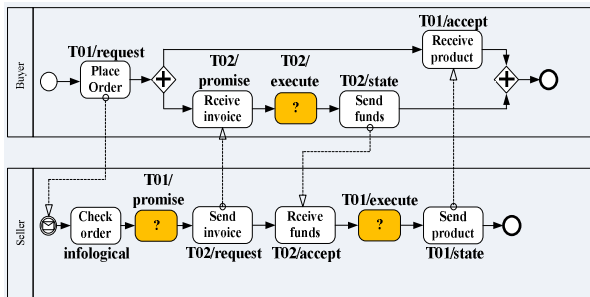


Figure 14. Identification of missing steps according to the DEMO process model (step 2).

In this example, the missing promise act (T01/promise) means the seller is implicitly accepting the buyer's request. Such decision may either be a design decision or the result of an incomplete specification. In either case, such an implicit promise makes this BPMN model to become ambiguous because it is not clear when the order is actually being accepted by the seller. It may be the case that the seller commits to selling the product to the buyer immediately after checking the order. But it may also be the case that the seller is only committed to the transaction if he sends the invoice. This makes the responsibility of the actors involved in the transaction to be ambiguous. For instance, if the buyer places an order and the seller does not send the invoice in a timely manner (e.g. as defined on a ser-

vice-level agreement), the model does not make explicit if the seller was already committed to send the invoice or not. The revised process makes such decision clear.

However, it is important to note that the revised process may not be ideal because there are usually several variations that are consistent and complete. Thus, the revised process primarily serves as a means to emphasize problem areas and to point the business process architects to possible solutions. A definite solution can only be found after an iterative analysis and revision of the process model according to the overall architecture of the organization in such a way that it suits the needs of its stakeholders and the overall architectural alignment parameters.

## 5 Conclusions

This paper presents a method to analyse the consistency and completeness of a process model with business transactions as defined in DEMO. Business transactions specify the pattern-based behaviour that describes how actors collaborate in order to achieve business results. The method takes as input a process model that is converted to a transactional model. The transactional model is then analysed and revised so that all transactions comply with the  $\Psi$ -theory axioms. Finally, the original BPMN process model is revised to become consistent with the transactional model and complete in the sense it expresses all transactional steps. The method was illustrated through the analysis and revision of a simple business process. We have also used this method to analyse a set of detailed business processes comprising several hundred activities at a large public legal institution. The method identified a set of implicit and missing actions from the original processes and prompted their revision so that the areas of responsibility and the contracts between the organizational actors became clear.

Nevertheless, the successful application of this method implies a sound understanding of the original process as well as working closely with its stakeholders so that it can be iteratively revised. This also means each step could end up be repeated several times until we get to the point where the input model satisfies the stakeholder's needs and is consistent and complete at the same time. Despite such shortcomings, such analysis will raise multiple questions about the contents of the original process models. The main question that tends to surface is about the actual meaning of the activities specified in the process model. Answering these questions contributes to reducing the ambiguity of the original specification and to clarifying implicit responsibilities and even determine to whom the activity is being delegated and the implications of such delegation on the

overall process. For example, on a department that handles customer complaints, all replies are ultimately the responsibility of that department, even if such responsibility is fully delegated or transferred to another actor. However, the transfer of responsibilities through delegation has impact on accountability and security. This means that a process model should clarify the areas of responsibility so that if a delegated actor fails fulfilling its goals, then the chain of accountability is explicit. We believe the method proposed in this paper is a contribution to this complex undertaking.

## 6 Acknowledgements

This project is partially supported by the EU-FP7 grant 269940, project TIMBUS, "Digital Preservation for Timeless Business Processes and Services".

## 7 References

- [1] Nuffel, D., Mulder, H., and Kervel, S., "Enhancing the Formal Foundations of BPMN by Enterprise Ontology", *Advances in enterprise engineering III - Lecture Notes in Business Information Processing*, 34(3), 2009, pp. 115-129.
- [2] Dijkmana, M., Dumas, M., and Ouyang, C., "Semantics and Analysis of Business Process Models in BPMN", *Information and Software Technology*, 50(12), 2008, pp. 1281-1294.
- [3] Zimmermann, O., Schlimm, N., Waller, G., and Pestel, M., "Analysis and Design Techniques for Service-Oriented Development and Integration", *INFORMATIK 2005, Jahrestagung der Gesellschaft für Informatik*, 2005.
- [4] OMG, *Business Process Model and Notation (BPMN), Version 2.0*, 2011.
- [5] Barjis, J., "Automatic Business Process Analysis and Simulation Based on DEMO", *Enterprise Information Systems*, Taylor & Francis, 1(4), 2007.
- [6] Recker, J., Indulska, M., and Rosemann, M., "How Good Is BPMN Really? Insights from Theory and Practice", *Proceedings of the 14th European Conference on Information Systems*, 2006.
- [7] Dietz, J., *Enterprise Ontology: Theory and Methodology*, Springer, New York, 2006.
- [8] Ettema, R., and Dietz, J., "ArchiMate and DEMO - Mates to Date?", *Advances in enterprise engineering III - Lecture Notes in Business Information Processing*, 34(4), 2009, pp. 172-186.
- [9] Nuffel, D.V., Huysmans, P., Bellens, D., and Ven, K., "Translating Ontological Business Transactions into Evolvable Information Systems", *International Conference on Software Engineering Advances*, 2010, pp. 58-63.
- [10] Dignum, F., and Dietz, J., "Communication Modelling: The Language/Action Perspective", *Second International Workshop on Communication Modeling (LAP 97)*, 1997.
- [11] Searle, J., *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, 1969.
- [12] Halpin, T., "Object-Role Modeling Version 2 (ORM 2)", in (Liu, L., and Özsu, M.T., 'eds.'): *Encyclopedia of Database Systems*, Springer, 2009.
- [13] Carlsen, S., "Comprehensible Business Process Models for Process Improvement and Process Support", *Advances on Information Systems Engineering, 8th International Conference, CAISE'96*, 1996.
- [14] Op't Land, M., Zwitzer, H., Ensink, P., and Lebel, Q., "Towards a Fast Enterprise Ontology Based Method for Post Merger Integration", *Proceedings of the 2009 ACM Symposium on Applied Computing (Enterprise Engineering track)*, 2009, pp. 245-252.
- [15] Op't Land, M., Proper, E., Waage, M., Cloo, J., and Steghuis, C., *Enterprise Architecture: Creating Value by Informed Governance*, Springer-Verlag, Heidelberg, Germany, 2009.
- [16] Op't Land, M., Middeljans, K., and Buller, V., "Enterprise Ontology Based Application Portfolio Rationalization at Rijkswaterstaat", *Via Nova Architectura*, 2008.