An Application of the ψ-theory to the Analysis of Business Process Models*

Artur Caetano[1,2,3], Aurélio Assis[1,3], José Borbinha[1,2], José Tribolet[1,3]

[1] Instituto Superior Técnico, Technical University of Lisbon,
Avenida Rovisco Pais 1, 1049-001 Lisboa, Portugal.
[2] Information Systems Group, INESC-ID, Lisboa, Portugal.
[3] Centre for Organizational Design and Engineering, INESC INOV, Lisboa, Portugal.
{artur.caetano, aurelio.assis, jose.borbinha, jose.tribolet}@ist.utl.pt

Abstract. This paper presents a method to analyse the consistency and completeness of process models according to the principles of the ψ-theory and the underlying concept of business transaction. Transactions specify the collaborative behaviour between actors while services are being requested and provided. The method assesses the consistency of a 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 and complete according to the transactional pattern. This enables to identify the problems on the original process model and to prompt areas of improvement.

Keywords. business process modelling, business transaction, ψ-theory, DEMO, BPMN.

1. Introduction

Business process modelling techniques can be used to analyse and communicate inter- and intra-organizational business processes as well as to develop business support information systems [1-5]. The project reported in this paper is motivated by the need to analyse the consistency of business process models. This capability is lacking in process modelling techniques since their focus is on the constructs required to define the models and not on methods to assess the actual quality of the models. Moreover, the lack of semantics behind process modelling languages and the unclear specification of their constructs contributes to the design of ambiguous models [6, 7]. An example of such ambiguity is the usage of natural language as the means to specify a process. These problems become evident when models or views need to be integrated or when the services to support a business process need to be consistently identified. On the other hand, methods to design business processes mainly focus on aligning business with technological concepts and do not prescribe the principles to design consistent business process [6]. The combination of these factors creates the need to define techniques to design new business processes or to provide the means to analyse existing process models with the goal of continuously improving their consistency while responding to the stakeholder's needs.

To address this problem, this paper uses the principles set forth by the ψ-theory to analyse the consistency of business processes. We consider a process model to be *consistent* if its activities

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

comply with the business transaction pattern. This pattern, which is part of the ψ-theory, defines communication steps that a requester and a producer perform while responding to a request. If a process complies with the transaction pattern then its specification describes who is responsible for the execution of its activities and why each activity is being performed. However, a business process can be consistent but still be missing activities required to fully produce the intended transactional results. This means that a *complete* process must not only be consistent but also specify all transactional activities. Thus, a process model that is both consistent and complete is a process that fully specifies an end-to-end collaboration pattern between a service requester and a service provider.

The next section of the paper introduces the ψ-theory along with the concept of business transaction. Section 3 describes the method to analyse business processes along with a running example. Finally, section 4 concludes the paper.

2. The ψ -theory

The ψ-theory (ψ is pronounced as PSI, standing for Performance in Social Interaction) is a theory about the ontological essence of organizations. It clarifies and explains the construction and operation of organizations. The operating principle of enterprises is that the employees, together with representatives of the customers and the suppliers, enter into and comply with commitments regarding the products that they cooperatively produce. This very basic understanding makes enterprises primarily social systems, of which the elements are human beings in their role of social individuals, bestowed with appropriate authority and bearing the corresponding responsibility. The ψ-theory is rooted in speech act theory [8], in social action theory [9, 10], and in information systems theory [11]. It is extensively discussed in [12]. The ψ-theory provides an effective notion of Enterprise Ontology, defined as the fully implementation independent understanding of the essence of an enterprise's organization. The Design & Engineering Methodology for Organizations [12] is a methodology for the engineering and implementation of organizations that is formally grounded on the ψ-theory . 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, 13]. DEMO can also deliver models which can be formally assessed and executed on DEMO automata [14]. DEMO can be used to assess existing BPMN models with the purpose of verifying their consistency [1]. The main contribution is thus combining the representational capability of the BPMN standard with the formal correctness of the DEMO principles. 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 [15]. The ψ-theory consists of four axioms [1, 12]. We briefly describe two of these axioms due to their significance to the method put forward in this paper.

2.1 The Operation Axiom

The operation axiom states that the operation of an enterprise results from the performance of actor roles. An *actor role* is an element of *authority* and *responsibility* which is fulfilled by a subject. A subject performs two kinds of acts: *production acts* (P-acts) and *coordination acts* (C-acts). By performing P-acts, subjects contribute to bringing about the function of the organization. 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 accomplished and made explicit. An 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.

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 package with the order items (the corresponding P-fact).

2.2 The Transaction Axiom

The transaction axiom states that C-acts and P-acts occur in a particular pattern and consists of an *order* followed by a *result* conversation. A conversation is a sequence of coordination acts between two actor roles required to achieve a result. A transaction evolves in three phases: the order phase (O-phase) where the request is made, the execution phase (E-phase) that produces the 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* or *requestor*. The role that executes the request, produces the fact and responds to the initiator is the *executor* or *provider*. The basic transaction pattern consists of the following steps:

1. Order phase
   - 1.1. The initiator makes a *request* (rq) to the executor to produce a specific P-fact. This P-fact will be the result of the transaction.
   - 1.2. 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.
2. Execution phase: the executor produces the P-fact as promised.
3. Result phase
   - 3.1. The executor *states* (st) to the initiator that the result is complete.
   - 3.2. The initiator *accepts* (ac) the result, thus successfully ending the transaction.

The basic transaction pattern assumes that all conversations have a successful outcome. DEMO also defines a standard transaction pattern that extends the basic pattern with four additional cancellation patterns. The standard transaction pattern is considered complete as it covers all possible communicative and rollback actions [12].

3. The Analysis Method

This section describes an iterative method to analyse the consistency and completeness of business process models. A business process model is *consistent* if the order of its actions complies with the transactional axiom. A business process model is *complete* if all transactional pattern steps can be mapped to its activities. Therefore, a process that is 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 consistency applies to any level of detail of the process. Therefore, if a business process is functionally decomposed then each level must also comply with the transactional axiom.

Throughout this paper we will consider the input model to be a BPMN process diagram, although other transformational modelling techniques could be used [16]. Thus, a process is specified as a flow of named activities that are performed by actors. The method comprises five steps. The first steps identify the production and coordination acts embedded in the input process model. The next steps assess the completeness and consistency of the process model according to the ψ-theory axioms. The input process model can then be reviewed according to the assessment. This enables the process model to be incrementally analysed and reviewed. The remainder of this section describes these steps along with an example of application.

3.1 Analyse the input process model (step 1)

The method starts with a BPMN process model and produces a DEMO process structure diagram that abstracts the coordination and production acts depicted on the input 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. 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. 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.

Consider the following examples: an activity that sends an electronic message is classified as a coordination act since it involves communication between actors and as a *forma* act because it represents a source uttering a message to a recipient. An activity that archives that message is classified as *forma* but it is a production act because it is generating a new production fact (the archived message). 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.

Next, each *performa* coordination act is classified as a *request*, *promise*, *state*, *accept* coordination act according to the transaction pattern. Using the standard transaction pattern implies further identifying the *decline*, *reject*, *stop*, *quit* acts. . 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.

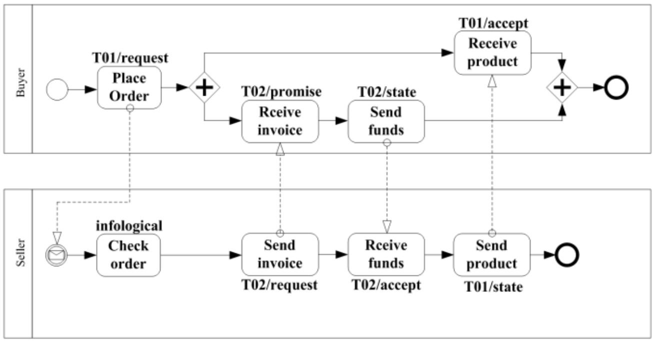

Figure 1. Input business process model after application of steps 1.

Figure 1 depicts a business process represented in BPMN describing the selling of a product to a buyer. The buyer places an order to a seller who checks its validity. Then the seller sends an invoice to the buyer. After receiving the payment the seller sends the buyer the product. This process is intentionally being specified at a high level of functional detail and it excludes error conditions and compensation actions due to space limitations.

The application of step 1 classifies the activity Check Order as infological (*informa*) according to the distinction axiom. The remaining activities are ontological (*performa*) acts. 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 result is shown in Figure 1.

3.2 Generate the DEMO models (step 2)

This step takes the classifications produced in step 1 and generates the corresponding DEMO process structure diagram. 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 DEMO process model.

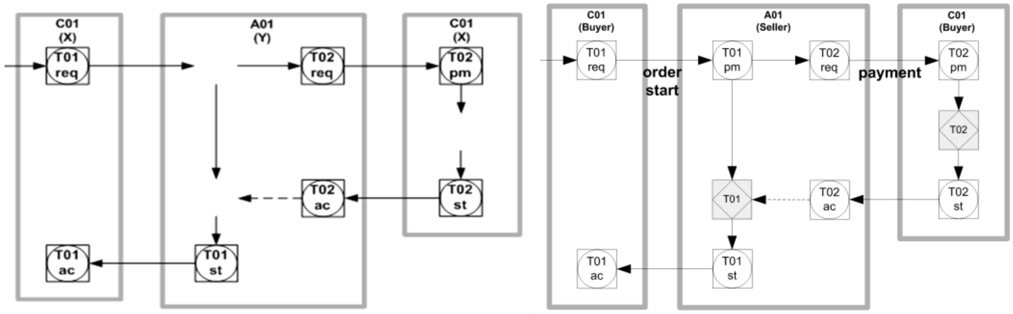

Figure 2. Incomplete process structure diagram after step 2 (left) and revised model after step 3 (right).

Step 2 of the bottom-up phase derives a DEMO process from the labelled business process model. The initial resulting model is shown in the left-hand side of Figure 2. It can be observed that this model is incomplete as several acts are missing.

3.3 Revise the DEMO models (step 3)

This step adds acts found to be missing during the previous step to the process model. These acts are the ontological acts that are not described explicitly in the input business process. The model is revised according to the transaction axiom so that each transaction goes through all the steps defined in the O-, E-, and R-phases. Next, the application of the composition axiom ensures that all of the transaction steps follow a logical sequence according to the pattern. The results of step 3 are process structure diagram (process model) and an actor-transaction diagram (construction model).

The result is the process model shown in in the right-hand side of Figure 2. This step has added the acts T01/pm, T01, T02 according to the transaction pattern. This revised model is now complete and consistent with the Ψ-theory axioms.

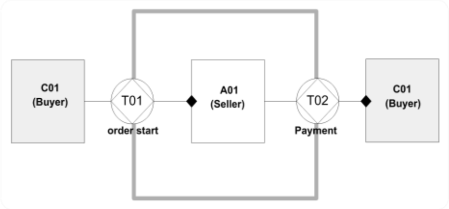

Figure 3. Construction model.

The actor-transaction diagram (ATD) depicted in Figure 3 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.

3.4 Revise the input process model (step 4)

This step performs a gap analysis that identifies the ontological acts in the revised DEMO model that are missing from the input business process model. Based on the results, the original BPMN process model is revised so that its activities match the missing ontological acts. The revised process becomes ψ-theory compliant, meaning it is now consistent with the transaction pattern and complete as it contains all transaction steps.

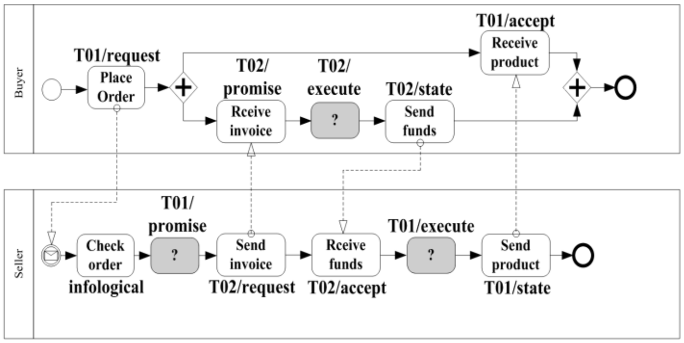

Figure 4. Identification of missing steps according to the transaction pattern (step 4).

This step identifies the missing activities on the original process model that derive from the missing acts. The revised process is depicted in Figure 4. The changes imply adding three activities, one for each missing ontological act: T01/promise, T01/execute and T02/execute. The name of these activities is actually irrelevant at this stage. The revised process is now consistent and complete with the ψ-theory axioms since all essential speech acts are explicit.

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 model 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 service-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 architects towards 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 principles.

3.5 Results of application

This method was applied to the analysis of two core business processes at a large organization (+2000 employees) that handles legal and judicial procedures. This was done using a software tool that fully automates parts of the method and supports the remaining semi-automated parts. The two processes total approximately 500 activities and cross 10 inter-organizational boundaries and involve more than 60 actor roles. The activities of these processes were already modelled and were detailed up to 4 or 5 levels of depth. The method was applied to these processes and the results were discussed and validated with the stakeholders. The application of the method took 20 person-days and was performed by a team process modelling experts who analysed the processes and met regularly with groups of stakeholders from the organization. Note that the example presented on the previous section is similar to this case study except in terms of complexity and extent. The following results were observed. The original processes lacked 25% of *production acts,* meaning that the original processes included activities that were not explicitly producing any tangible results. The following observations apply to the missing *coordination acts*:

- Missing 25% of request acts. Results are being produced without an explicit request being made and thus it is not possible to identify an accountable service initiator. This is particularly important in processes that are executed by multiple actors, especially inter-organizational actors.
- Missing 50% of promise acts. Requests are implicitly confirmed, often without any governing contract. The service performer starts producing a result without agreeing with the initiator and thus it is not formally accountable for the production.
- Missing 25% of state coordination acts. Production results are not explicitly communicated. The implication is that is not clear whether the responsibility of checking the completion of a result lies on the initiator or on the executor.
- Missing 40% of accept acts. Production results are not being accepted by the initiator. There is no formal acceptance and therefore the initiator does not confirm that the results comply with the request. Moreover, the executor cannot assess the quality of its results since there is no formal acceptance of the product.

4. Conclusions

This paper presents a method to analyse the consistency of a process model based on the ψ-theory. A business transaction specifies a pattern that describes how actors collaborate. The method takes as input a process model that is converted to a transactional model. The transactional model is then revised so that all transactions comply with the ψ -theory axioms. Finally, the original process model is revised to become consistent 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 institution. The method identified a set of implicit and missing acts 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 questions about the contents of the 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. This means that a process model should clarify the areas of responsibility and the chain of accountability. We believe the method proposed in this paper is a contribution to this complex undertaking.

References

1. Nuffel, D., H. Mulder, and S. Kervel, *Enhancing the Formal Foundations of BPMN by Enterprise Ontology.* LNBIP, Advances in enterprise engineering III, 2009. 34(3): p. 115-129.
2. Dijkmana, M., M. Dumas, and C. Ouyang, *Semantics and analysis of business process models in BPMN.* Information and Software Technology, 2008. 50(12): p. 1281-1294.
3. Zimmermann, O., N. Schlimm, G. Waller, and M. Pestel, *Analysis and Design Techniques for Service-Oriented Development and Integration,* in *INFORMATIK 2005,* 2005: Bonn, Germany.
4. Sousa, P., A. Caetano, A. Vasconcelos, C. Pereira, and J. Tribolet, *Enterprise architecture modeling with the UML 2.0*, in *Enterprise Modeling and Computing with UML*, P. Rittgen, Editor 2006, Idea Group Inc. p. 67-94.
5. Caetano, A., A. Rito Silva, and J. Tribolet. *A Role-Based Enterprise Architecture Framework.* In *24th Annual ACM Symposium on Applied Computing*. 2009. Hawaii, USA.
6. Barjis, J., *Automatic business process analysis and simulation based on DEMO.* Enterprise Information Systems, Taylor & Francis, 2007. 1(4).
7. Recker, J., M. Indulska, and M. Rosemann, *How Good is BPMN Really? Insights from Theory and Practice*, in *Proceedings of the 14th ECIS*, J. Ljungberg and M. Andersson, Editors, 2006: Goeteborg, Sweden.
8. Searle, J., *Speech Acts: An Essay in the Philosophy of Language.* 1969, Cambridge: Cambridge University Press.
9. Dignum, F. and J. Dietz, *Communication Modelling: The Language/Action Perspective*, in *Second International Workshop on Communication Modeling (LAP 97)*1997: The Netherlands.
10. Habermas, J., *The theory of communicative action - reason and the rationalization of society.* 1984, Boston, MA, USA: Beacon Press.
11. Langefors, B., *Information System Theory.* Information Systems, 1977. 2: p. 207-210.
12. Dietz, J., *Enterprise Ontology: Theory and Methodology.* 2006, New York: Springer. 244.
13. Ettema, R. and J. Dietz, *ArchiMate and DEMO - Mates to Date?* Advances in enterprise engineering III - Lecture Notes in Business Information Processing, 2009. 34(4): p. 172-186.
14. Nuffel, D.V., P. Huysmans, D. Bellens, and K. Ven, *Translating Ontological Business Transactions into Evolvable Information Systems,* in *International Conference on Software Engineering Advances* 2010: France p. 58-63.
15. OMG, *Business Process Model and Notation (BPMN), version 2.0.* 2011.
16. Carlsen, S., *Comprehensible Business Process Models for Process Improvement and Process Support*, in *Advances on Information Systems Engineering,* P. Constantopoulos, et al , Editors, 1996, Springer.