

A Toolkit For Business Process Owners to Capture Early System Requirements

Ken Decreus, Frederik Gailly, Geert Poels

Faculty of Economics and Business Administration, Ghent University
{ken.decreus, frederik.gailly, geert.poels}@ugent.be

Abstract. Semantic Business Process Management (SBPM) raises Business Process Management (BPM) from the IT level, where it mostly resides now, to the business level, where it belongs. SBPM provides a rich ontological description of both enterprise and process aspects, and aims to support business process modellers by means of SBPM modelling tools. Unfortunately, no explicit support is foreseen to capture early system requirements coming from the business process owner. To meet this need, we propose a toolkit approach and provide a mapping algorithm to semi-automatically insert the acquired business knowledge in the SBPM modelling environment.

Keywords: business process owner, early-phase requirements engineering, requirements elicitation toolkit, i* modelling framework, BPMO.

1 Introduction

Since the 1980s, information systems development methods have been extensively discussed in literature [1]. During the first phase of these development methods, i.e. requirements analysis, Conceptual Modelling (CM) techniques are frequently employed to capture the meaning of information by constructing computer-based symbol structures [2].

One of the current problems [3] in the field of CM is the gap between the work on enterprise ontologies (e.g. TOVE [4], Enterprise Ontology [5]) and the workflow-centric view on business processes (e.g. BPEL [6]). Enterprise ontologies describe the conceptual structures of an enterprise without considering how these models are executed in production systems. On the other hand, workflow-centric process representations capture business activity sequencing and other execution flow aspects in order to deploy processes in run-time software environments. However, this focus on the control flow within business processes makes them less suitable for accessing the business space at the conceptual (i.e. implementation-independent) level.

The ARIS tool and methodology [7] addressed this gap by combining a conceptual model of an enterprise with the actual production system. Semantic Business Process Management (SBPM) [3, 8] further extends this Business Process Management (BPM) view to increase the level of automation. Like in traditional BPM methodologies, the Semantic Business Process (SBP) life cycle has four phases: SBP modelling, SBP configuration, SBP execution and SBP analysis. When taking a closer

look at SBP modelling, we discover that actors playing the role of *business process modeller* are the envisioned users of the SBPM modelling tools [9]. One of the main tasks of business process modellers is to create process models using the Business Process Modelling Ontology (BPMO - version 1.4), and annotate these models with ontological constructs. Unfortunately, SBPM does not provide explicit support to capture the requirements coming from the *business process owner*, which normally serve as input for the work of the business process modeller. We will refer to these requirements as ‘early’ requirements as they relate to the goals of the business process, the (alternative) means to achieve these goals (i.e. activities and resources) and constraints that apply (e.g. limited resources, time constraints), but generally do not include detailed requirements regarding manual/technological solutions.

Our research suggests that explicit support to capture these early system requirements will improve communication between the process owner and the process modeller. We will propose a toolkit (Section 2), to be used by the process owner to facilitate requirements elicitation and structuring, and we formulate a semi-automatic translation (Section 3) from these early requirements into preliminary BPMO diagrams (Figure 1). It is our objective to reuse the entire SBPM ontology structure during this automatic translation, although in this paper we only focus on BPMO. Later on, the process modeller can further enrich the generated BPMO diagrams using available SBPM ontologies.

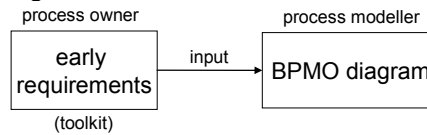


Fig. 1: Empowering business process owners

2 A toolkit for the process owner

Research in the area of Management Science suggests that product design done by the product user is far more efficient than innovation by product manufacturers [10]. It is proposed to outsource need-related innovation tasks to the users themselves after equipping them with *toolkits* for user innovation. For example, a toolkit to create pizzas at home could consist of a box full of pizza dough with tomato sauce (generic solution), while several toppings plus a manual describing tasty topping combinations would be provided (adapting solution to user needs). Toolkits allow rapidly changing user needs (e.g. changing a mozzarella pizza into a pepperoni by adding salami) and help to elicit tacit user requirements by user iteration (e.g. discovering a rare but tasty combination of toppings). As an example in the business/IT field, Ricken & Steinhorst [11] propose to empower a business user by considering the Supply-Chain Operations Reference-model (SCOR) as a toolkit for business process innovation.

In our research, we propose to use the first phases of the Tropos methodology [12] as a toolkit mechanism. The Tropos project provides a model-driven methodology where *i** models [13] are used to drive the generation of software systems. The first two phases are acquisition of early requirements and defining late requirements, on which subsequent design and implementation are based. Our interest in Tropos is

motivated by the large body of research done, and by the evidence of industry practices using Tropos as an information systems development method [14]. Although run-time environments such as TAOM4E [15] support the Tropos methodology, we limit the usage of Tropos concepts in this paper to the graphical notation.

The *i** modelling framework provides us with different modelling constructs to specify intentionality. A *goal* node in the goal tree shows that there are alternative ways of achieving the goal, but *no specific instructions* are given how to achieve the goal (e.g. when a car owner enters a repair shop and asks to “just get it fixed”). A *task* node shows that we *specifically* know what to do but there are *constraints* on how to do it (e.g. the car owner asks the repair shop to raise the engine idle settings in order to fix the engine). A *resource* node shows that getting the resource is unproblematic (e.g. getting new oil to fix the car), and a *softgoal* node will state the non-functional requirements to be attained while performing the task (e.g. have the car fixed economically).

Using these constructs, the *i** Strategic Dependency (SD) model describes a business process in terms of intentional dependency relationships among agents. The *i** Strategic Rationale (SR) model describes the internal process in more detail from the point of view of one of the agents. In the SR model, task-decomposition links provide a hierarchical description of intentional elements and means-ends links provide the understanding about why an actor would engage in some tasks, pursue a goal, need a resource or want a softgoal.

To give a practical example of how the toolkit can be used, we will describe the (simplified) use case ‘VOIP Order Fulfilment’ of Telekomunikacja Polska (TP) [16], related to the provisioning of the Voice-Over-IP (VOIP) telephony service. The TP business manager, responsible for this VOIP process, starts by plotting two actors (Customer and TP) and their mutual dependencies. The customer is dependent on TP to buy VOIP services, but is not expected to be interested in how this goal is fulfilled (“just give me VOIP”). Furthermore, the customer is expected to pay a certain amount of money to TP (Figure 2). Using task decomposition and means-ends links, the process owner expresses a high-level view on the current way of working: handling billing, handling orders and managing staff. Orders can be requested via telephone, in person or via an order page on the customer website (Figure 3). Suppose now that the process owner desires that the current order page on the corporate website, which has to be filled in manually, is replaced by an automated order system using Semantic BPMS technology.

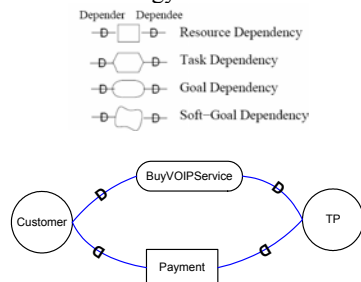


Fig. 2: SD model - early requirements phase

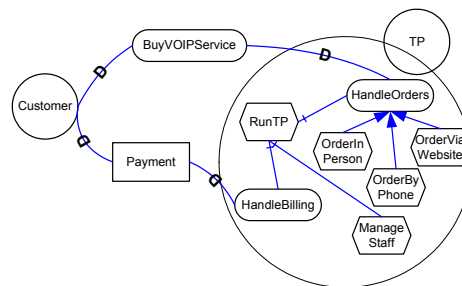


Fig. 3: SR model - early requirements phase

The second phase of the Tropos project, definition of late requirements, reworks these models. First, an actor to represent the software system-to-be is included in the original SD model, and then a means-end analysis for this ‘system actor’ is initiated. As a consequence, the process owner adds the system as actor (VOIP Fulfilment System) to the original, early phase SD model. He assumes that customers would like to place orders in a secure environment, while TP wants the system to automate the existing manual processing of orders (Figure 4). The VOIP fulfilment system has to manage orders by taking customer order requests and fulfilling these orders. As known to the process manager, the CustomerOrderRequest goal is decomposed in three sub-tasks: CustomerIdentification, CustomerVerification and CaseCreation. To ensure the security standards, CustomerIdentification and CustomerVerification are expected to contribute positively to the overall system security (Figure 5).

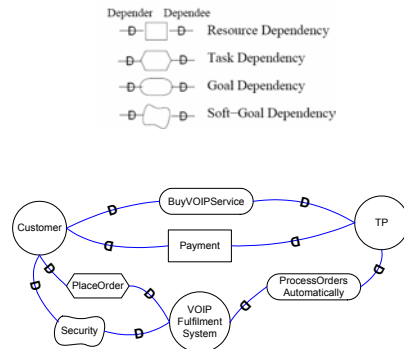


Fig. 4: SD model - late requirements phase

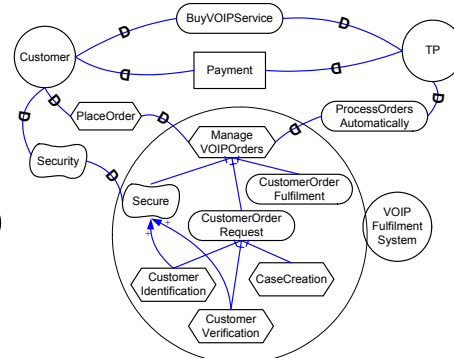


Fig. 5: SR model - late requirements phase

3 Mapping toolkit information to BPMP diagrams

When the process owner has structured his preliminary system requirements regarding the process, the information in the toolkit is ready to be transferred to the SBPM modelling environment. We will propose a mapping algorithm by means of the TP use case. A more formal version of the algorithm could be based on the formal specification of i^* , Formal Tropos [17], and can be included in future research.

The mapping algorithm starts by creating a BPMP diagram and by giving it the name of the highest i^* task found in the systems’ actor SR model (ManageVOIPOrders). Followed by adding a Start and End event, we investigate the task decomposition links leaving the ManageVOIPOrders node. When encountering an i^* Task leaf node, the algorithm inserts a Task element in the BPMP diagram. Nevertheless, the type of Task element is yet to be decided: when manual work is expressed by the i^* leaf node, create a ManualTask; when existing web services can be used to implement this leaf node, create a WebServiceTask; when no existing web services are known at this point, create a GoalTask in the BPMP diagram and express desired functionality by means of Web Service Modeling Ontology (WSMO) Goals.

Discovering the goal node CustomerOrderRequest, all task decomposition links leading to subtasks are investigated. The first task leaf node found is CustomerIdentification; thus, a GoalTask CustomerIdentification is added to the BPMO diagram (as no existing web services are known at this point). Using the same reasoning will add GoalTasks CustomerVerification and CaseCreation. The mapping algorithm will stop when discovering the goal CustomerOrderFulfilment as this goal has no children.

Finally, the process modeller observes the semi-automatically generated BPMO diagrams in the SBPM modelling environment, and can add further implementation details such as control flow aspects or additional GoalTasks such as CustomerIdentifiedProblems (Figure 6).

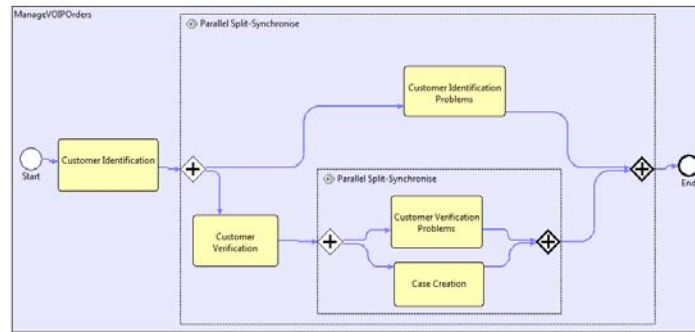


Figure 6: The BPMO diagram resulting from Tropos models

4 Conclusion & Future Research

In order to empower the business process owner to structure his early phase system requirements regarding the process, we proposed a toolkit based on the first stages of the Tropos project. A mapping algorithm was described to translate the toolkit knowledge into BPMO diagrams, as defined by the SBPM modelling environment.

Future research could explore how knowledge acquisition methodologies, other than the ones used in Tropos, can support early requirements elicitation. Furthermore, it would be possible to adapt an existing Tropos run-time environment (such as TAOM4E) to create a working demo for our toolkit. A formal version of the proposed mapping algorithm can also be designed based on the Formal Tropos language. Finally, the integration between the entire SBPM ontology structure and the toolkit can be intensified (e.g. Business Function Ontology).

5 References

- [1] N. Ahituv, M. Hadass, and S. Neumann, "A Flexible Approach to Information System Development," *MIS Quarterly*, vol. 8, pp. 69-78, Jun84 1984.
- [2] J. Mylopoulos, "Information modeling in the time of the revolution," *Information Systems*, vol. 23, pp. 127-155, 1998.
- [3] M. Hepp and D. Roman, "An ontology framework for semantic business process management," *Proceedings of Wirtschaftsinformatik*, Feb 28 - Mar 2 2007.
- [4] M. Fox and M. Gruninger, "Enterprise Modeling," *AI Magazine*, vol. Fall 1998, pp. 109-121, 1998.
- [5] J. Dietz, *Enterprise Ontology: Theory and Methodology*: Springer-Verlag New York, Inc., 2006.
- [6] OASIS, "Web Services Business Process Execution Language Version 2.0," <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, 2007.
- [7] A.-W. Scheer, *ARIS - Business Process Modeling*: Springer-Verlag New York, Inc., 2000.
- [8] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel, "Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management," in *Proceedings of the IEEE International Conference on e-Business Engineering*: IEEE Computer Society, 2005.
- [9] SUPER, "D5.1: Semantic Process Modelling Environment," in www.super-ip.org, 2007.
- [10] E. v. Hippel and R. Katz, "Shifting Innovation to Users via Toolkits," *Management Science*, vol. 48, pp. 821-833, 2002.
- [11] A. Ricken and A. Steinhorst, "Why working with reference models increases process innovation," *BPTrends*, vol. February, 2006.
- [12] J. Castro, M. Kolp, and J. Mylopoulos, "Towards requirements-driven information systems engineering: the Tropos project," *Information Systems*, vol. 27, pp. 365-389, 2002.
- [13] E. S.-K. Yu, "Modelling strategic relationships for process reengineering," University of Toronto, 1995, p. 181.
- [14] H. Estrada, A. Rebollar, O. Pastor, and J. Mylopoulos, "An Empirical Evaluation of the i* Framework in a Model-Based Software Generation Environment," in *Advanced Information Systems Engineering*, 2006, pp. 513-527.
- [15] A. Perini, A. Susi, L. Penserini, D. Bertolini, and A. Novikau, "TAOM4E," <http://sra.itc.it/tools/taom4e/>, 2008.
- [16] SUPER, "D3.2: Dynamic Composition Reasoning Framework and Prototype," in www.super-ip.org, 2007.
- [17] A. Fuxman, J. Mylopoulos, M. Pistore, and P. Traverso, "Model Checking Early Requirements Specifications in Tropos," in *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE '01)*: IEEE Computer Society, 2001.