

A Goal-Oriented Requirements Engineering Method for Business Processes

Ken Decreus, Geert Poels

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

Abstract. The field of requirements engineering (RE) for business processes has grown during the last several years. As business processes are needed to fulfil organizational goals, the information captured in goal models provides a basis for designing business processes. Although research has started to explore how to transform goal models into business process models, current transformation methods need further research. This paper proposes a tool-supported method to model goals as part of the business requirements for business processes and to automatically generate business process design skeletons that respond to these business requirements.

Keywords: Goal-Oriented Requirements Engineering, Business Process Modelling, Business-Strategy Context Process, Atlas Transformation Language

1 Introduction

The information needs of an organisation set requirements for its information systems and the setting of goals, and the formulation of business strategies to achieve these goals, leads to business requirements for the business processes of the organisation. We define *business requirements* for business processes as the overall set of requirements that relate to business processes as given by the Business Motivation Model (BMM) [1] of OMG, such as vision, mission, goal, strategy, objective and tactic. More specifically, a *vision* describes the future state of the enterprise, without regard to how it is to be achieved, and *mission* indicates the ongoing activity that makes the vision a reality. A *goal* indicates what must be satisfied on a continuing basis to effectively attain the vision, and a *strategy* is a long-term activity designed to achieve a goal. An *objective* is a specific and measurable statement of intent whose achievement supports a goal, and a *tactic* is a short-term action designed to achieve an objective.

In a business process-centred organization, the architectural view on implementing business requirements through Business Process Management Systems (BPMS) [2] is given by Fig. 1. On the top layer, called Strategy Thinking Layer, artefacts such as vision, mission, goal, strategy, objective and tactic are positioned. The layer below is the Business Process Architecture Layer, where the business process models that document the business processes reside. The third layer is the Business Process Execution Layer, where BPMS-executable versions of the business process models on the layer above are managed in order to run the business (by means of automated or

human activities). The bottom layer, called Business Process Infrastructure Layer, contains the IT infrastructural services (e.g. web services, service-oriented software applications) that are used to automate the non-human parts of business processes.

The importance of implementing requirements by means of BPMS software is illustrated by Gartner Research [3], which estimates that by 2015 30% of business applications will be developed by means of BPMS technology. As traditional software packages are expected to play a less important role, we foresee a growing need for RE techniques that are adapted to BPMS packages.

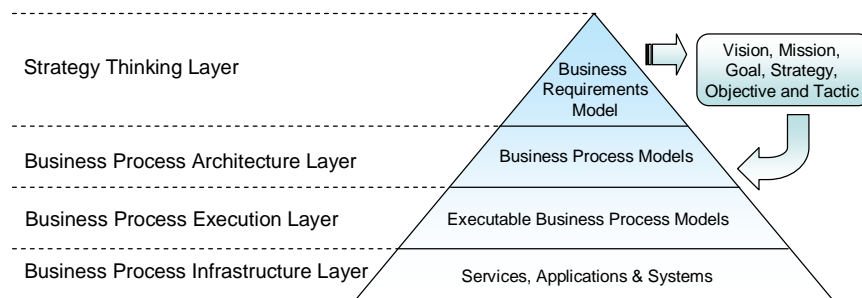


Fig. 1. Implementing business requirements through a Business Process Management System

Our research intends to contribute to the realization of the Business Process Architecture Layer. This paper presents an approach to model Strategy Thinking Layer goals as part of the business requirements for business processes and to automatically generate business process design skeletons (captured in models on the Business Process Architecture Layer) that respond to these business requirements. Our first contribution is taking an existing goal-oriented requirements engineering method, called the B-SCP (Business-Strategy Context Process) method [4], to create the Business Requirements Model of an organisation. The unique value proposition of B-SCP is combining the i* goal modelling language [5], Jackson's Problem Frames [6] and Role Activity Diagrams [7] into one overall top-down method. We extended the B-SCP method by developing a graphical editor for visually creating business requirement models and to generate B-SCP models based on the existing metamodels. Our second contribution is offering automatic transformation mappings to create business process design skeletons (in Business Process Modelling Notation - BPMN [8]) out of the B-SCP models. To this end, we reuse the work of Lapouchnian et al. [9] to support the generation of business process models.

The target user of our approach is a domain expert (called 'business user') who works in a business process-centred organisation and understands both high-level strategy concepts (such as business goals) and low-level operational details (such as the way business processes are organized). The development of this approach is based on the working hypothesis that it is useful and valuable to first model new or changed goals in a business requirement model and next to generate business process design skeletons out of this business requirement model in such a way that the changes to business process designs, needed to comply with the new/changed requirements, can

be done more easily/effectively (compared to directly changing existing business process models).

2 Goal-Oriented Requirements Engineering for Business Processes

Our method provides the business user with an Eclipse-based Business Requirements Editor, which he/she employs to design a new Business Requirements Model (to initiate the Strategy Thinking Layer) or to adapt an existing Business Requirements Model (e.g. to add a new or modified business process to the Business Process Architecture Layer). In this paper, we will illustrate the first scenario that is detailed below (Step 1 to Step 6), and of which the first three steps are based on the B-SCP method of Bleistein et al. [4] and the fourth step relates to the work of Lapouchnian et al. [9]. The resulting Business Requirements Model is a hierarchical model of context diagrams, that have corresponding requirement diagrams. For instance, Fig. 2 shows the Business Requirements Model for the Seven-Eleven Japan (SEJ) case study [10], that describes the information-based strategies that have helped SEJ become a top performing retailer in Japan, selling high quality products through an industry-wide supply chain network of manufacturers, distributors, third-party logistics providers and franchise shops.



Fig. 2. Eclipse-based Visual Business Requirements Editor

Step 1. Identify the Business Model Participants and Their Relationships (B-SCP Original). The organisation of interest is determined, together with other business model participants (such as customers, allies and suppliers) and the flows of money, product and information between the participants are identified. For instance, SEJ is the organisation of interest, that relates to customers, franchise stores, combined delivery centres and suppliers (Fig. 2 – Context Diagram DA).

Step 2. Identify the Top-Level Business Requirements (B-SCP [4] Original). The VMOST (Vision, Mission, Objectives, Strategy, Tactics) method is used to deconstruct the motivational aspects of a company into business requirements, and the rules of OMG's Business Motivation Model (BMM) [1] are employed to relate the discovered business requirements. For instance, the vision 'create a chain of SEJ convenience stores where you can find a solution for any of your daily life problems at hours when needed' is supported by the mission 'Use IT to coordinate a supply chain of business partners' (Fig. 2 – Requirement Diagram RA).

Step 3. Identify Business Process Participants and Their Relationships (B-SCP [4] Original). For each business process that the business user wants to elaborate, determine the business process participants and the flow of products or information between the participants. The scope of the business process is defined by identifying the main business process participant, and by selecting other business process participants in function of the main one. For instance, the Point of Sale system of a franchise store relates to a product, the clerk and a customer (Fig. 2 – Context Diagram DB).

Step 4. Refine the Top-Level Business Requirements (Adapted from B-SCP [4]). Given a specific context of business process participants and their relationships, the top-level business requirements (such as vision, mission, goal, strategy, objective and tactic) should be refined into business requirements for business processes (such as the required business process itself, required subprocesses and required tasks). The original B-SCP framework considers all kinds of business requirements for business processes as *tactics*. In contrast, we consider a *business process* as something that realizes a *tactic* by means of an ordered collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer [11]. An activity in a business process could be a *subprocess*, which is a business process included in a higher level business process, or a *task*, which is an atomic activity performed by an end-user and/or an application. For instance, the business process 'Customer Checkout' has a subprocess 'Payment Process', that consists of a task 'Pay with cash' (Fig. 2 – Requirement Diagram RB).

Step 5. Add Control Flow Annotations (Based on Lapouchnian et al. [9]). In this paper, we want to support the basic control-flow patterns as published by the Workflow Patterns Initiative (WCP-1: Sequence, WCP-2: Parallel Split, WCP-3: Synchronisation, WCP-4: Exclusive Choice, WCP-5: Simple Merge) [12]. To this end, we let the business user annotate the links in the Business Requirements Model with control flow annotations. More specifically, the business user can choose means-end links, sequential AND decomposition links, parallel AND decomposition links, and OR decomposition links. Firstly, means-end links indicate a relationship between an end (i.e. *i** soft/hard goal) and a means (i.e. *i** task) for attaining this end (e.g. the mission is a means for attaining the vision). Secondly, a sequential AND decomposition link defines that the execution of an *i** node depends on the left-to-right sequential execution of *all* nodes indicated by the link (e.g. the customer checkout process is achieved when the clerk starts by taking the products presented

for purchase and ends by giving the receipt). Thirdly, a parallel AND decomposition link defines that the execution of an *i** node depends on the execution of *all* nodes indicated by the link without imposing a particular order (e.g. to assess a customer, the clerk has both to assess customer age and gender, but the order in which this is done doesn't matter). Fourthly, an OR decomposition link defines that the execution of an *i** node depends on the execution of *at least one* node indicated by the link (e.g. customers can pay the entire amount with cash or VISA, or partially cash and partially VISA).

Step 6. Transform Selected Problem Diagram into Business Process Model. The business user selects a problem diagram (e.g. Fig. 2 – Problem Diagram B consisting of RB and DB) from the Business Requirements Model, and activates the automated transformation mappings via the Eclipse environment. The automated mappings are implemented by means of the Atlas Transformation Language (ATL) project [13]. The basic requirements to run an ATL project are having a source metamodel, a target metamodel and an instance of the source metamodel. Based on the defined ATL mappings, an instance of the target metamodel will be generated from the source instance.

An extract of our ATL mappings that we defined is shown in Fig. 3. Firstly, sequential routing of business process activities, that are related to a domain of interest, map to workflow control-flow pattern WCP-1 (Fig. 3 – rule 4). Secondly, parallel routing of business process activities, that are related to a domain of interest, map to workflow control-flow pattern WCP-2 and WCP-3 (Fig. 3 –rule 5 and 6). Thirdly, conditional routing of business process activities, that are related to a domain of interest, map to workflow control-flow pattern WCP-4 and WCP-5 (Fig. 3 –rule 7 and 8). The output of the transformation mappings is a serialized business process model following the BPMN notation, which can be manually refined in an Eclipse-based BPMN editor (Fig. 4).

```
(1) rule ProblemDiagram{
  from a : BRM!ProblemDiagram
  to b : BPMN!BpmnDiagram(name <- a.name)}

(2) rule DomainOfInterest{
  from a : BRM!DomainOfInterest
  to b : BPMN!Pool(name <- a.name),
  startevent : BPMN!Activity(activityType <- 'EventStartEmpty'),
  endevent : BPMN!Activity(activityType <- 'EventEndEmpty'),
  firstSequence : BPMN!SequenceEdge}

(3) rule Task{
  from a : BRM!Task
  to b : BPMN!Activity(activityType <- 'Task', name <- a.name)}

(4) rule ANDDecomposition_Sequence{ --Implementation of WCP-1
  from a : BRM!ANDDecomposition(self.type = #SequentialOrder)
  to b : BPMN!SequenceEdge(id <- 'Sequence Edge')}

(5) rule ANDDecomposition_Parallel_FirstOccurrence{ --Implementation of WCP-2 and WCP-3
  from a : BRM!ANDDecomposition(self.type = #ParallelOrder and
  BRM!ANDDecomposition.allInstances()->first())
  to b : BPMN!Activity(activityType <- 'GatewayParallel'),
  c : BPMN!SequenceEdge(id <- 'Left Parallel Edge'),
  d : BPMN!SequenceEdge(id <- 'Right Parallel Edge'),
  e : BPMN!Activity(activityType <- 'GatewayParallel'),
  f : BPMN!SequenceEdge(id <- 'Edge Closing Parallel Construction')}

(6) rule ANDDecomposition_Parallel_OtherOccurrences{ --Implementation of WCP-2 and WCP-3
  from a : BRM!ANDDecomposition(self.type = #ParallelOrder and not
  BRM!ANDDecomposition.allInstances()->first())
```

```

to b : BPMN!SequenceEdge(id <- 'Left Parallel Edge'),
c : BPMN!SequenceEdge(id <- 'Right Parallel Edge'})

(7) rule ORDecomposition_FirstOccurrence{ --Implementation of WCP-4 and WCP-5
from a : BRM!ORDecomposition(BRM!ORDecomposition.allInstances()->first())
to b : BPMN!Activity(activityType <- 'GatewayDataBasedExclusive'),
c : BPMN!SequenceEdge(id <- 'Left Conditional Edge'),
d : BPMN!SequenceEdge(id <- 'Right Conditional Edge'),
e : BPMN!Activity(activityType <- 'GatewayDataBasedExclusive'),
f : BPMN!SequenceEdge(id <- 'Edge Closing Conditional Construction')}

(8) rule ORDecomposition_OtherOccurrences{ --Implementation of WCP-4 and WCP-5
from a : BRM!ORDecomposition(not BRM!ORDecomposition.allInstances()->first())
to b : BPMN!SequenceEdge(id <- 'Left Conditional Edge'),
c : BPMN!SequenceEdge(id <- 'Right Conditional Edge')}

```

Fig. 3. ATL extract

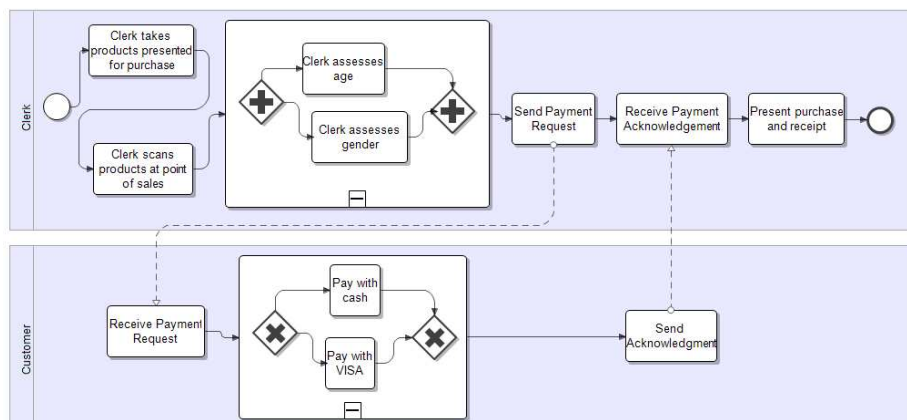


Fig. 4. Generated Business Process Model

3 Discussion

The overall goal of our research is to reduce the existing gap between RE research and industrial RE practice [14]. When considering the current research on goal-oriented requirements engineering, a lot of research is done related to the i^* goal language. Nevertheless, few published studies exist on applying the i^* goal language into practice, and indications exist that practitioners of large-scale industrial projects are unable to understand i^* models well enough to validate the requirements of the system they were building [15]. As the B-SCP framework was proposed to address the known shortcomings of i^* and to leverage the existing knowledge of Jackson's Problem Frames, we considered the B-SCP framework as the starting point of our work.

The differentiation between a business requirements language and a business process language is the result of a deliberate design choice. As a modelling language is always conceived with a certain purpose in mind [16], we believe it is easier to represent goals and business processes using different languages, and to provide automatic translations between these languages, instead of choosing one modelling language to represent both goals and business process concepts. With low modelling complexity (e.g. modelling one clearly understood business process), creating a Business Requirements Model could be seen as an overhead cost. But, as real-world

business process modelling projects often quickly grow in complexity, business users can use the Business Requirements Model as an overview (or one could say, an overarching strategically aligned Business Process Architecture), and automatically generate as much business process models from the Business Requirements Model as they require.

Finally, we want to discuss the difference between ‘business requirements for business processes’ and ‘software requirements’. In terms of Fig. 1, business requirements are to be situated in the Strategy Thinking Layer, and relate to the motivational aspects of a company. Based on the Business Requirements Model of the Strategy Thinking Layer, this paper proposes a method to build the Business Process Architecture Layer by generate business process models that describe all kinds of activities (performed by machines or humans). Typically, these business process models are used by requirement engineers to discover software requirements [17] such as functional requirements, non-functional requirements, constraints, interfaces, etc. In contrast, in the context of a BPMS, these business process models do not act as documentation but could be executed -after adding the necessary run-time components- in the BPMS. So in this paper, the notion of ‘software requirements’ coincides with the business requirements for non-human, automated business processes.

4 Conclusion and Future Work

This paper presents an approach to model Strategy Thinking Layer artefacts as part of the business requirements for business processes and to automatically generate business process design skeletons (captured in models on the Business Process Architecture Layer) that respond to these business requirements. Our first contribution is extending an existing goal-oriented requirements engineering method, called the Business-Strategy Context Process (B-SCP) method, such that it can be used to create the Business Requirement Model for an organisation. Our second contribution is offering automatic transformation mappings to create business process design skeletons out of business requirements models. The expected implications of our research is to empower a (non-technical) business user with a serialized, yet intuitive way to represent his business requirements, going from strategies and goals till business processes and activities, and to allow these business users to generate business process models from these requirements. This should allow the business user to provide technical experts with reusable IT assets instead of providing merely paper-based requirements that requires more interpretation from (non-business) technical experts.

The current limitations of our work are the limited support for workflow control-flow patterns (only WCP-1 until WCP-5), and the lack of full-scale validation. In order to get initial feedback on the use and perceived value of our method, we decided to conduct a number of small-scale pilot studies in a specific context (Policy Modelling), in order to evaluate and refine our solution before considering larger-scale and more general case study research. The main finding [18] of our pilot studies was the need for a thorough preparation of the participants in understanding our definitions, tools and method steps. For instance, a correct understanding among the participants should be reached about what is a mission, strategy, tactic, business

process, or task, as participants could have different interpretations. The full-scale validation should check whether the newly added activities in Step 4 could work well and contribute to the production of artefacts of higher quality in the latter steps. Next, we need to discuss the quality of the resulting BPMN artefact, and the applicability of transformation such that our transformation technique could be used for other goal-oriented RE techniques.

References

1. OMG: Business Motivation Model - <http://www.omg.org/spec/BMM/>. (2009)
2. Shaw, D.R., Holland, C.P., Kawalek, P., Snowdon, B., Warboys, B.: Elements of a business process management system: theory and practice. *BPMJ* **13** (2007) 91-107
3. Woods, J., Genovese, Y.: Delivery of Commodity Business Applications in a BPMS Does Not Mean You Should Customize the Applications. Gartner Research **G00139084** (2006)
4. Bleistein, S.J., Cox, K., Verner, J., Phalp, K.T.: B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. *Information and Software Technology* **48** (2006) 846-868
5. Yu, E.S.-K.: Modelling strategic relationships for process reengineering (PhD Thesis). University of Toronto (1995)
6. Jackson, M.: Problem Frames: Analysing and Structuring Software Development Problems. Addison-Wesley, New York (2000)
7. Ould, M.A.: Business Processes: Modelling and Analysis for Reengineering and Improvement. Wiley, Chichester, New York (1995)
8. OMG: BPMN 1.2 specification - <http://www.omg.org/spec/BPMN/1.2/>. (2009)
9. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. *Business Process Management* (2007) 246-261
10. Bensaou, B.M., Uchino, H., Mitani, K., Noishiki, M.: Seven-Eleven Japan: Managing a Networked Organisation. INSEAD Euro-Asia Centre - Case Study 05/97-4690 (1997)
11. Hammer, M., Champy, J.: Reengineering the corporation : a manifesto for business revolution. HarperBusiness, New York (1994)
12. Russell, N., Hofstede, A.H.M.t., Aalst, W.M.P.v.d., Mulyar, N.: Workflow Control-Flow Patterns : A Revised View. BPM Center Report BPM-06-22, BPMcenter.org (2006)
13. Eclipse: ATL - ATLAS Transformation Language - <http://www.eclipse.org/m2m/atl/>.
14. Kaindl, H., Brinkkemper, S., Bubenko Jr, J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., Leite, J.C.S.d.P., Mead, N.R., Mylopoulos, J., Siddiqi, J.: Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. *Requirements Engineering* **7** (2002) 113-123
15. Maiden, N.A.M., Jones, S.V., Manning, S., Greenwood, J., Renou, L.: Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study. *Advanced Information Systems Engineering* (2004) 3-21
16. Mylopoulos, J.: Information modeling in the time of the revolution. *Information Systems* **23** (1998) 127-155
17. IEEE: Recommended Practice for Software Requirements Specifications - IEEE Standard 830-1998. (1998)
18. Decreus, K., Poels, G., Kharbili, M.E., Pulvermueller, E.: Policy-Enabled Goal-Oriented Requirements Engineering for Semantic Business Process Management (To Be Published). *International Journal of Intelligent Systems* (2009)