

## Chapter 3.12

# Enterprise Specific BPM Languages and Tools

**Steen Brahe**  
*Danske Bank, Denmark*

### ABSTRACT

Many enterprises use their own domain concepts when they model business processes. They may also use technology in specialized ways when they implement the business processes in a Business Process Management (BPM) system. In contrast, BPM tools often provide a standard business process modeling language, a standard implementation technology and a fixed transformation that may generate the implementation from the model. This makes the tools inflexible and difficult to use. This chapter presents another approach. It applies the basic model driven development principles of direct representation and automation to BPM tools through a tool experiment in Danske Bank. We develop BPM tools that capture Danske Banks specific modeling concepts and use of technology and which automate the generation of

code. An empirical evaluation reveals remarkable improvements in development productivity and code quality. We conclude that BPM tools should provide flexibility to allow customization to the specific needs of an enterprise.

### INTRODUCTION

Business Process Management (BPM) is currently receiving much focus from the industry. Top management demands to understand and control their business processes and agility to adjust them when market conditions change. This can be achieved through Process Aware Information Systems (Dumas et al., 2005). A BPM system (Jablonski and Bussler, 1996; Leymann and Roller, 2000) is one example of such a system. It allows execution and automation of a business process that can be described explicitly as an executable workflow.

DOI: 10.4018/978-1-60566-669-3.ch002

Although the hype about BPM and process automation is high, previous work has shown that it is relatively complex to understand, model and implement a business process as an executable workflow (Brahe, 2007). First the process must be understood, second it must be formalized and modeled at a highly conceptual and logical level, and third the process design must be transferred to technology. Many software vendors have complete BPM tool suites for modeling and implementing business processes. Such tools are mostly based on a predefined process modeling language like the Business Process Modeling Notation (BPMN) (White, 2006) for capturing the business process at the conceptual level and one technology like the Business Process Execution language (BPEL) (BPEL, 2003) for implementing the process. These tools also assume a fixed development process where only two models exist, i.e. the conceptual business process and the implementation.

Using such tools causes two challenges for an enterprise that has specific requirements to its development process, uses its own modeling concepts and uses technology in specialized ways; First, a standardized modeling notation does not allow users to use domain concepts and may contain too many modeling constructs which makes the tool difficult to use. The models may also be difficult to understand and use as a communication media. Second, transformation of a model into implementation must be done manually as the enterprise may use a variety of technologies to implement the process and not only e.g. BPEL as many state-of-the-art tools support today. Even if one technology as e.g. BPEL is used, the enterprise may be using its own implementation patterns which cannot be generated because the transformations are hard-coded into the BPM tools.

The approach behind current BPM tools is similar to the extinct Computer Aided Software Engineering (CASE) tools from the 90es. They also often used a standard modeling language, one implementation technology and a standardized transformation. Their limited flexibility in

supporting enterprise specific standards was one of the reasons why they were never accepted (Windsor, 1986; Flynn et al., 1995).

This chapter takes another approach than state-of-the-art BPM tools. In order to avoid the CASE trap we must come up with an approach that allows an enterprise to use its own modeling notations and specific use of technology. Our hypothesis is that this can be achieved through applying the basic model driven development (Stahl et al., 2006) principles of direct representation and automation (Booch et al., 2004) to BPM tools; An enterprise should be able to model its business processes directly in enterprise specific concepts, decide on a target platform and write transformations that encapsulate its specific use of technology, and that automate the generation of code.

This leads us to the research question which we will answer through this chapter: Does an enterprise specific BPM tool improve the efficiency and quality of modeling and implementing business processes, how difficult is it to create, and is it worth the effort?.

We use a design research approach to answer this question; We will implement above hypothesis through an experiment where we develop BPM languages, tools and transformations for a specific enterprise. Successively, these will be empirically evaluated to show the validity of the hypothesis. We use Danske Bank, the second largest financial institute in northern Europe, as a case study. In lack of sufficient industrial standards, Danske Bank has defined its own development process and uses a number of different tools to support it. This has caused several challenges as described by Brahe (2007).

A prototype tool was developed to show that it provide value to develop BPM tools fitted for the needs of a specific enterprise. The prototype illustrates that it is possible to do model driven development of a business process with nearly 100% code generation. The prototype is fitted specially for Danske Banks development process and consists of three different Domain Specific

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/enterprise-specific-bpm-languages-tools/44102](http://www.igi-global.com/chapter/enterprise-specific-bpm-languages-tools/44102)

## Related Content

---

### A Decision Support System for ERP Implementation in Small and Medium-Sized Enterprises

Mahmood Ali, Ying Xie and Joanna Cullinane (2013). *Sociotechnical Enterprise Information Systems Design and Integration* (pp. 97-121).

[www.irma-international.org/chapter/decision-support-system-erp-implementation/75877](http://www.irma-international.org/chapter/decision-support-system-erp-implementation/75877)

### Evolution and Maintenance of Web Sites: A Product Line Model

Rafael Capilla and Juan C. Duenas (2005). *Managing Corporate Information Systems Evolution and Maintenance* (pp. 255-271).

[www.irma-international.org/chapter/evolution-maintenance-web-sites/25752](http://www.irma-international.org/chapter/evolution-maintenance-web-sites/25752)

### Applying the ONTOMETRIC Method to Measure the Suitability of Ontologies

Asunción Gomez-Perez and Adolfo Lozano-Tello (2005). *Business Systems Analysis with Ontologies* (pp. 249-269).

[www.irma-international.org/chapter/applying-ontometric-method-measure-suitability/6125](http://www.irma-international.org/chapter/applying-ontometric-method-measure-suitability/6125)

### Outsourcing Non-Core Business Processes: An Exploratory Study

Adriana Romaniello and B. Dawn Medlin (2009). *Selected Readings on Information Technology and Business Systems Management* (pp. 499-515).

[www.irma-international.org/chapter/outsourcing-non-core-business-processes/28656](http://www.irma-international.org/chapter/outsourcing-non-core-business-processes/28656)

### Enterprise Resource Planning Acceptance Model (ERPAM): Extended TAM for ERP Systems in Operational Phase of ERP Lifecycle

Simona Sternad and Samo Bobek (2012). *Measuring Organizational Information Systems Success: New Technologies and Practices* (pp. 179-204).

[www.irma-international.org/chapter/enterprise-resource-planning-acceptance-model/63453](http://www.irma-international.org/chapter/enterprise-resource-planning-acceptance-model/63453)