A Metamodel Based Perspective on the Adaptation of a Semantic Business Process Modeling Language to the Financial Sector

Burkhard Weiß
University of Muenster,
European Research Center for Information Systems (ERCIS)
Leonardo-Campus 3,
48149 Münster, Germany
burkhard.weiss@ercis.uni-muenster.de

Axel Winkelmann
University of Muenster,
European Research Center for Information Systems (ERCIS)
Leonardo-Campus 3,
48149 Münster, Germany
axel.winkelmann@ercis.uni-muenster.de

Abstract

Process modeling is an important prerequisite to process reorganization and management. As a result, many companies have spent much effort on process documentation, while hardly gaining equivalent benefits in the analysis and usage of the resulting process models. To balance the cost-benefit-ratio of process modeling projects with respect to their later usage, especially regarding automatic process model analysis (i.e. for optimization purposes etc.), new domain-specific and thus semantic business process modeling languages (SBPML) have been proposed for selected domains. In this paper we investigate the adaptability of SBPML from the public sector to the banking sector from the perspective of the language’s metamodel, since banks are currently highly involved in modeling initiatives to industrialize and optimize their process landscapes and are unsatisfied with existing modeling approaches regarding the cost-benefit-ratio of modeling. While taking a metamodel perspective on the language artifact itself, we derive requirements for process modeling from the domain of financial institutions and present findings on the adaptation of SBPML, giving a complete conceptual model of the SBPML method for banks.

1. Introduction

Process modeling is an important prerequisite to process reorganization and management in all institutions and enterprises. It has been a major issue for at least the last two decades with roots going back to the beginning of the twentieth century. Business process modeling can be seen as a way of capturing the implicit process knowledge of an organization and document it explicitly in a (semi-)formal way. As a result, many companies have created literally hundreds of meters of “wallpaper” with process models. However, with such an effort spent on process documentation, there are hardly equivalent benefits in the analysis and usage of process models [1]. In a recent study, we asked business process management (BPM) experts from 38 banks about their satisfaction with process modeling and analysis possibilities of the process modeling approaches used in their represented banks (cf. Figure 1). Less than 32% were very satisfied or satisfied with the economic efficiency of process model creation and less than 27% were very satisfied or satisfied with the possibilities of automating process model analysis (e.g., for business process reengineering purposes). Regarding the overall satisfaction with the cost-benefit ratio of process modeling, none of the respondents was very satisfied and less than 37% were satisfied.

Figure 1. Satisfaction with business process modeling methods in banks

The widely used general purpose modeling languages such as the Business Process Modeling Notation (BPMN), the Event-Driven Process Chain (EPC) or Flow Chart Diagrams consist of syntactic elements without any predefined semantic domain
information. Most languages only allow for the unstructured annotation of semantic domain information during modeling. As a result, the languages do not sufficiently offer automated semantic analysis possibilities. Of course, it is possible to count the usage of certain IT systems or the switching from one department to another within the same process. However, it is very difficult and many times not possible to automatically identify, e.g. media breaks or redundant document checks within a process model or even across process models. In many cases, highly trained advisors with sufficient domain expertise are necessary to create new models and to evaluate the inherent syntax and semantics of existing models.

In contrast, semantic business process modeling languages allow for predefined elements of the domain language in order to make statements about the problem domain. Hence, this predefinition of semantic elements enables an easier modeling [2], even for non-experts, coupled with an automated analysis of the resulting process models [3]. However, with inherent descriptive elements, concerning the specific domain, the modeling language is bound to one domain only and is of limited value to other domains. For example, process activities that are common in retail companies (such as replenishment) are not common in the service sectors. In this article, we discuss the following research question by analyzing domain requirements and providing and comparing the metamodels of a semantic business process modeling language (SBPML) for public administrations (originally called PICTURE [2,4]) and for banks: “Is it possible to adapt a semantic process modeling language from one domain to another and if so, how does the metamodel for the language look like and how does it change according to different domain requirements?”

We apply our research question to a SBPML that was originally developed for process modeling and analysis in the public sector, but is currently transferred to the financial sector.

2. Theoretical framing

2.1. Business process modeling languages

Process modeling languages are used for the documentation of a system behavior or its structure. Many of today’s modeling languages reach back to Petri’s proposal in 1962 [5], where he proposes the representation of the system behavior with the help of states and transitions. Such languages – frequently also referred to as business process modeling languages – are employed e.g. for the description of activities within software development projects or for the documentation of work activities in an enterprise. Examples of these are BPMN, EPC, UML Activity Diagrams or IDEF3 that focus on the representation of corporate processes.

The differentiation between syntactic and semantic business process modeling languages draws upon the question whether a language is generic / domain-independent or domain-specific [6]. Typically, in a syntactic modeling language the meaning of most elements is not defined with regard to a particular business domain. In contrast, in a semantic modeling language most elements are defined with regard to a particular domain and are not subject to individual modeling. For instance, predefined activities such as “receive an application”, “formal assessment”, or “send out notification” are understood by experts in the public sector, but are not necessarily understood and used the same way by retailers. Hence, most of the vocabulary of a semantic modeling language has its origin in the specific vocabulary of the domain. As the distinction between syntactic and semantic modeling languages is not always clear-cut, Pfeiffer [6] proposes the differentiation with regard to the amount of domain-related elements. Consequently, languages such as BPMN, EPC, UML Activity Diagrams and IDEF3 are syntactic languages with regard to most domains.

Over the past years, the information systems (IS) communities have started discussing semantic business process modeling languages [7]. These discussions have especially evolved around automated semantic analysis desires concerning process models. These analyses are linked to the specific domain and address aspects that are relevant to managers such as process quality, substantial process weaknesses, savings through process automation etc.

To date, there are various research projects and prototypes that deal with pattern identification and semantic annotation within semi-formal process models [8]. For instance, Thom [9] identifies typical block activity patterns as business functions frequently found in business processes. Iochpe et al. [10] discuss a suite for business processes based on the reuse of context-sensitive workflow patterns. Often, process modeling languages are linked to ontologies. For example, Lin [11] introduces an ontology-based semantic annotation approach to enrich and reconcile semantics of process models. Thomas and Fellmann [12] also use metadata to connect actual process models to ontologies. Those approaches need a domain ontology and a (manual) matching between business models and ontological concepts (e.g. the Semantic Business Process Modeling Notation – SPBMN [13]). In our point of view, this two-step approach is very difficult to communicate and use in practice. Hence, a
The research methodology for this article follows the design science paradigm [22] that deals with the rigorous construction of scientific artifacts like methods, languages, models, and implementations with regard to specific research goals. The artifacts to be constructed have to represent an innovative contribution to the existing knowledge base within the actual research discipline. Subsequent to the construction of the artifacts, these have to be evaluated in order to prove their fulfillment of the research goals.

In this contribution the scientific artifact is the semantic business process modeling approach and its underlying metamodel. This artifact aims at solving the relevant problem of modeling and automatically analyzing business processes in banks (cf. Section 2). The approach presented here (cf. Section 3 for the existing modeling approach and its views) makes an innovative contribution to the existing knowledge base once it has been adapted to the needs of banks (cf. Section 4). In order to compare the differences between the modeling language variant used for the public sector and that used for the financial sector, the metamodel is introduced and compared (cf. Section 5). Finally, we reflect our research with respect to a conclusion, limitations and the contribution and give an outlook (cf. Section 6).

3. Domain-specific semantic business process modeling language

The domain-specific SBPML is a semantic process modeling approach and is based on the concept of semantic building-block based languages (SBBL) [6]. It has been instantiated for the domains of public administrations [4, 23] and is currently being adapted to that of banks. Like other process modeling approaches (e.g. ARIS EPC [24]) it uses the concept of views to structure relevant information and thereby reduce complexity. In the language there are four different views to capture different aspects relevant for the documentation and analysis of business processes. These views comprise a process view, a business object view, an organizational view and a resource view.

The core constructs of this language are domain-specific process building blocks (PBB), which have an integrating role by connecting all views (e.g. the IT system [resource view], used for executing an activity in the form of the PBB “Enter Data into IT”, is linked to the PBB just as the credit application [business object view] is used to specify which document is handled during the PBB). A PBB represents a certain set of activities within a process and applies a domain-
specific vocabulary. PBBs are atomic, have a well-defined level of abstraction that clearly defines their application scope and differentiates them from other PBBs, and are semantically specified by a domain concept. With PBBs problems like naming conflicts during model comparison are avoided, because the name of a PBB is specified by the language designer rather than the modeler. Examples for PBBs are “Document / Information Comes In”, “Perform a Formal Verification”, “Enter Data into IT”, or “Archive Document”.

PBBs belong to the process view and represent the lowest abstraction level of a process model. They are contained within different variants of subprocesses. The subprocesses, representing activities of just one organizational unit, are in turn part of a larger process, which usually involves multiple organizational units and thus multiple subprocesses. The process view describes in detail how a process is executed. It also acts as the integrating view for the other views. The process model does not only document which activities have to be performed in which sequence but also who performs them (organizational view), what is processed by them (business object view) and whereby the activity is supported (resource view).

Additional facts about the processes can be collected with the help of attributes assigned to each PBB. Attributes specify the properties of the corresponding PBBs in detail. For example, a possible attribute for the PBB “Enter Data into IT” is “Duration”. Attributes provide the core information for a subsequent process analysis. They establish a connection to the business object, organizational, and resource view.

The organizational view captures details about the organizational context of processes. The organizational model uses the elements of organizational units, positions and staffing of positions to build a hierarchical organization plan. The elements from this view can be used in the process models to depict who is carrying out certain tasks and who is responsible for a process.

The business object view contains details about the information which is processed in the course of a business process. The business objects model structures different types of business objects like a credit application or a credit agreement. This allows depicting input and output of processes and enables the analysis of the information flow throughout process landscapes.

The resource view contains information about the resources which are needed during a process. Resources include software and hardware but also certain skills or knowledge required for a task. The resource model structures these elements and allows for a consistent view on them. The annotation of resources used within processes enables analyzing IT support and resource consumption of processes.

4. Deriving domain-specific requirements for process modeling in the financial sector

Confronted with the goal of transferring a SBPML to the application domain of banks, a multi-method triangulational approach (conducting a literature review, expert interviews, and case studies) was chosen to assess the feasibility of adapting the initial modeling language artifact. This was done by cooperating with a German and a Russian bank in 2008 and 2009.

The German bank was a specialized bank, which focused solely on sales, production and steering of instalment credits. It operated these as a single product for over 900 partner banks, which in turn offered these consumer credits to their customers. The bank was operating in Germany and Austria with an additional 60 subsidiary credit shops in different cities. It employed over 2,460 people in 2008, who together as a bank served 443,000 customers, totaling a credit volume of 4.9 billion euros. As the credit process is the most studied process in literature and this bank was only focusing on it, this bank seemed to be a well-suited starting point for us.

Complementing this specialized bank, we chose a universal bank from a newly industrialised country, namely Russia, thus differing in both bank type and location from the first research partner bank. The bank offered a wide range of typical banking products to its customers, including cash services, credits, deposits, cards and payments. The banking activities were spread over multiple regional branch offices. The bank served 37,000 small and medium enterprises in South Russia, had over 160,000 depositors and issued credits in an overall volume of 94.2 billion roubles in 2008. As of January 1st 2009, the bank employed over 2,000 people in 132 subsidiary offices in South Russia.

By cooperating with these two banks it was possible to derive a series of domain-specific requirements for process modeling in banks through a series of interviews, round-table discussions and workshops. During these extensive cooperations the banks provided process models of varying scope and size in the EPC notation, in order to assess if the same content could also be expressed in the original SBPML notation. Our domain-specific process modeling requirements (cf. Figure 2), derived from several evolutionary modeling iterations with over 261 processes modeled in both banks – comprising 418 subprocesses, 1071 subprocess variants and 3,590 activities, are summarized in the following.
The process building blocks, encapsulating the actual semantics of the domain had to be changed slightly. In several evolutionary modeling iterations with over 261 processes modeled in both banks – using the PBB selection approach and criteria (e.g. usefulness, soundness, significance, completeness etc.) as suggested by [25], we were able to verify if the existing PBBs inherent to public administrations also worked for banks. We found that the PBBs had to be change in part. From originally 24 PBBs from the public administration sector 8 were merged into 4 PBBs (e.g. “Make / Receive Payment” with a differentiating attribute specifying if a payment was made or received), 1 PBB was eliminated (e.g. “Record / Register”) as it was not used in banks and 5 PBBs were added (e.g. “Make Accounting Transaction”) as these described frequent activities in banks. A peculiarity we found, was the desire of the banks to also document their system activities to prevail this knowledge for the future. Therefore one of the additional PBBs we created was “System Activity”, which could be used to describe activities carried out automatically and in the background of IT applications, but which were nevertheless worthwhile documenting on an aggregated level to enhance understanding of the subprocess flows. Nevertheless, as the description of system activities was a requirement from the bank’s IT department, we had to make a tradeoff, as the detailed modeling of system activities cannot sufficiently be done through the introduction and application of a single PBB, but will need other modeling approaches designed for specifically designing and documenting IT systems (e.g. UML).

The semantic attributes (e.g. the necessity of a four-eyes-principle during execution of a specific activity etc.) on all levels (PBBS, variants, subprocesses and processes) had to be adjusted largely, as the banks not only used a different vocabulary compared to that of public administrations, but also preferred to document as much detail as possible within their process models for later analysis purposes. The original SBPML specification included 163 attributes. During the course of our investigation we had to change a number of attributes (11), remove several attributes specific to public administrations (17) and add new general as well as bank specific attributes (149). The resulting attribute set comprised 304 analyzable attributes.

Customer activities as well as communication with the customer had to be documented in the process models. In opposition to the public administration sector, banks demanded to model their processes also on the customer side to optimize these and in order to assess their process convenience to customers. Thus, a new organisational role for external partners had to be introduced into the organizational view to allocate certain process flows to the role of a customer. Therefore, the role of the customer was introduced and replaced the former citizen role for communication activities in process flows with regard to banks.

The control flow paradigm had to be extended to be able to describe the full complexity of activity flows in large process models with various different subprocesses and subprocess variants. While the original SBPML only allowed to model sequences of subprocesses, disregarding from which subprocess variant the control flow would be passed on to which other subprocess variant(s) of another subprocess, the banks demanded to also model these sequences of subprocess variants due to auditing requirements to check for legal compliance. According to the original SBPML specification it was not possible to connect different variants from different subprocesses with each other. A restriction in the form that variant A and variant B of subprocess II only followed after variant A from subprocess I thus could not be depicted (cf. Figure 2). It was only possible to connect a complete subprocess with another complete subprocess and then differentiate in how many cases a variant of the following subprocess was actually carried out. This issue was resolved by changing the previous control flow paradigm, so that modeler A owning subprocess A could establish an outgoing control flow from each variant of his owned subprocess to a succeeding subprocess. The following subprocess, owned by modeler B, could then use these multiple incoming control flows and connect them to individual variants of the following subprocess, while using percentage values to specify the likelihood of each variant’s occurrence after a previous subprocess (cf. Figure 2). By means of these percentage values it was now also possible to implicitly model AND-, XOR- and OR- connectors without actually having explicit language artifacts for these connectors and thus keeping SBPML simple and intuitive to users without an IS background.

Due to frequent in- and outsourcing initiatives in banks, the banks also demanded to add a new abstraction level to the process view, where subprocesses could be grouped to form value creating subprocess bundles (cf. Figure 3). These groups would represent relatively autarkic economic services, that could be offered to other banks as stand-alone services in the future.
The idea was to recombine these value creating (sub)process bundles under a white-labeling approach with different subprocesses from external bank partners, since not all external bank partners may want to outsource their complete crediting (sub)processes to our given partner bank, but desire to selectively outsource only those subprocesses, which they think our studied bank can do better (e.g. in terms of costs, quality, cycle time or even related to the management of the involved risks). In addition, the banks demanded that outsourced subprocesses should be depicted as black box external processes for better process model comprehension.

Finally, the banks were used to a top-level view upon their process landscape in terms of a process map (cf. Figure 2). This depicted the overall internal process value chain, including the most important processes of the given bank. Its purpose was to give the bank’s management and third parties an intuitive overview of how the bank’s processes interacted and where their position within the internal value chain of the bank’s business was located.

With these various changes and extensions of the original SBPML it was not only possible to describe all process flows of both banks, but – more importantly – it was possible to semantically encapsulate as much process knowledge as appropriate, while enabling the banks to use the newly created process models for further additional purposes (e.g. semi-automatic weakness analysis etc.) [14].

5. Comparison of domain-specific SBPML metamodels

Recapitulating our original research question, we have found that there are a number of domain-specific requirements in banks that differ from public administrations. Thus the original SBPML must first be adapted in the previously mentioned ways to fulfill the demands of banks regarding business process modeling. To answer the second part of our research question on how these changes to the original specification reflect on the artifact’s metamodel, we
have developed a metamodel of the resulting SBPML for banks (cf. Figure 3). In this metamodel we have indicated the changes in contrast to the original specification that were necessary, to meet all process modeling requirements of the banks. For a better comprehension, we will go through all parts of the metamodel by explaining the different views in detail, while highlighting the changes made.

With regard to the process view we had to add a new construct to enable the modeling of process maps. A process map (new entity type) is made up of one or multiple business areas, which can be used to group the processes of a bank on the highest aggregation level. Each process can be put into a process sequence (new relationship type) to depict the core value chain of the process landscape. The original constructs to depict a value chain network (eliminated relationship type), consisting of multiple processes (e.g. process “issue business license” and process “issue real estate agent license”) within a value chain network context (eliminated entity type), such as in a citizen lifecycle model, where the “citizen wants to open a business and become a real estate agent” were not needed in banks, as these aspects could be modeled more appropriately with the help of the new construct of a process map.

Each process (entity type) results in a product or service (entity type) and is either a core process (entity type) or a support process (entity type). While core processes are stand-alone processes that are not “used” by other processes, support processes are typically used by one or more processes. Each process is owned by an organizational unit and comprises several subprocesses (entity type) that are operated by different organizational units in a bank. These subprocesses can be grouped to value creating subprocess bundles (new entity type) for the purpose of gaining an intermediate but aggregated view on complex processes in the context of white-labeling or outsourcing initiatives. Every subprocess has one or more variants (entity type/interpreted relationship type) that can be connected to other variants through the concept of a variant sequence (new relationship type) of the same or other processes, thus enabling complex control flows.

Therefore, the old concept of subprocess sequences (eliminated relationship type) becomes obsolete. Each subprocess variant contains process building blocks (reinterpreted relationship type) that follow a strict sequential flow with the help of the construct process building block sequence (relationship type). Since the same process building blocks are part of several alternative paths in a subprocess, called variants, there is also a construct to depict especially twin process building blocks (used in multiple variants of the same subprocess) on the subprocess level (entity type).

Each process building block used for modeling is of a certain domain-specific process building block type (entity type). All process building block types reflect the domain-specific activities that are frequently used within a domain – here that of banks. Each process building block type has domain-specific process building block type attributes (entity type). Correspondingly, each process building block also has domain-specific process building block attributes including their values (entity type). If a subprocess triggers and also requires input from another support process or subprocess, this fact is described using parallel processes (entity type). As mentioned earlier, the process building block integrates all other views – especially through the various attributes that can be used to specify and instantiate a process building block type. It can have multiple (model) relations (reinterpreted entity type) to other related elements (entity type), while specifying the semantics of each depicted relation type (entity type). An example illustrating this fact can be the process building block “death certificate arrives”, which is of the process building block type “Document / Information Comes In”. This activity is related to the document “death certificate” (object view) and the death certificate can be sent from a governmental institution (organizational view) to the bank – thus depicting the relation between the bank and the governmental institution.

From the organizational view each activity, depicted in a process building block, is carried out by an activity operator (changed entity type). This activity operator has a specific position within the hierarchy of the organizational unit (entity type) and is of a certain job position type (entity type) that classifies it either as an operating or executing job position type. For each job position type, within the organizational unit hierarchy, there is a specified instantiation in the form of a planned job position (entity type) that can be owned by a certain employee that owns the position type (changed relationship type). Furthermore, each job position type can be part of a board (e.g. board of directors) (entity type) that can be defined within an organization. A board or organizational unit are both an organizational element (entity type) that can be a communication partner in a process activity. In addition, also external partners (entity type), such as customers (changed entity type), business companies (entity type) and governmental institutions (entity type), can be communication partners. To be able to depict activities that are executed by customers, these activities can also be owned by customers (changed entity type). Finally, also subprocesses can be executed by any type of external partner, which also makes it possible to document outsourced business subprocesses as black boxes.
Figure 3. SBPML metamodel indicating adaptations and extensions of the original SBPML for banks' domain requirements

The business object view and the resource view did not require any changes. A business object in both domains was either an information (entity type), a document (entity type) or a material object (entity type).
the conceptual metamodel changed less and less, the documentation of the processes [26-28]. In addition, as using the available PBB attributes for a systematic added in a standardized and analyzable fashion by also the existing process models, additional details could be reexpressed using the SBPML notation. As opposed to notations in the bank (e.g. EPC) could be fully previously described in other process modeling within these cases. We found that all processes, and universal bank and also successfully been tested engineered to the specific requirements of a specialized domain-specific SBPML for banks that has been defined and presented the underlying metamodel of a banks.

However, we only studied two in-depth cases in the banking sector to derive our domain-specific requirements and transfer the SBPML to the new domain of banks, wherefore our metamodel and resulting SBPML for banks may not yet be complete, even though we assume that at this stage it may well also serve other business process modeling projects in banks.

Thus, from a contribution point of view, we have defined and presented the underlying metamodel of a domain-specific SBPML for banks that has been engineered to the specific requirements of a specialized and universal bank and also successfully been tested within these cases. We found that all processes, previously described in other process modeling notations in the bank (e.g. EPC) could be fully reexpressed using the SBPML notation. As opposed to the existing process models, additional details could be added in a standardized and analyzable fashion by also using the available PBB attributes for a systematic documentation of the processes [26-28]. In addition, as the conceptual metamodel changed less and less, the

6. Conclusion, limitations, contribution and outlook

Originally, we wanted to address the question of if it was possible to adapt a domain-specific process modeling language to a new domain and which changes would be required to the specification of the SBPML – especially from a metamodel perspective. We found that the transfer of a domain-specific language to another domain is possible, but partial changes and extensions have to be made to the metamodel of the investigated SBPML to implement the domain-specific requirements. In our case we conclude that the effort of adapting was fair, since both domains use a similar vocabulary and due to their administrative nature both have similar activities within their business processes. Thus only a few conceptual changes on the metamodel level and several semantic changes on the instantiation level of the metamodel (e.g. PBBs and attributes) had to be made. However, we only studied two in-depth cases in the banking sector to derive our domain-specific requirements and transfer the SBPML to the new domain of banks, wherefore our metamodel and resulting SBPML for banks may not yet be complete, even though we assume that at this stage it may well also serve other business process modeling projects in banks.

With respect to an outlook, we suggest further research on the feasibility of the new SBPML for banks in different institutions in the financial sector. From our research experience we can imagine that the developed artifact will also work well in closely related domains and institution types, such as insurance companies, juridical institutions (such as law courts or lawyers), tax consultancies and even business consultancies [30]. Therefore, we think that these new domains present interesting vistas for future studies on domain-specific requirements.

Nevertheless, our two case studies have been limited in their scope, wherefore this first derived proposition will still need further validation effort by means of further insight from further process modeling case studies in the financial sector. Also – with regards to limitations of the approach presented here – we must conclude that our results present a first language design effort that may be subject to further extensions. Specifically, different banking requirements and different contextual situations (e.g. use of the method in different departments from a bank, other than those analyzed in our case studies, or for extended usages like business process compliance or operational risk management) could lead to further changes in the language and the presented results / language extensions present tradeoffs between what seems to be generalizable and useful for many banks (e.g. modeling of customer activities, the introduction of the higher abstraction level of process maps, a more detailed control flow concept etc.) and what may be of lower importance (e.g. detailed modeling of IT system activities, for which more suitable modeling language like UML already exist).

With respect to an outlook, we suggest further research on the feasibility of the new SBPML for banks in different institutions in the financial sector. From our research experience we can imagine that the developed artifact will also work well in closely related domains and institution types, such as insurance companies, juridical institutions (such as law courts or lawyers), tax consultancies and even business consultancies [30]. Therefore, we think that these new domains present interesting vistas for future studies on domain-specific semantic business process modeling languages and their application.

7. References