Business Modeling Experience for a State Pension Voluntary Insurance Case

Alcedo Coenen[1], Bas van Gils[2], Charlotte Bouvy[3], Roel Kerkhofs[3],
and Sander Meijer[4]

[1] Sociale Verzekeringsbank, Amstelveen, The Netherlands
acoenen@svb.nl
[2] BiZZdesign, Enschede, The Netherlands
b.vangils@bizzdesign.nl
[3] O&i - Partners in BPM, Utrecht, The Netherlands
{c.bouvy,r.kerkhofs}@oi.nl
[4] Verdonck, Klooster & Associates, Zoetermeer, The Netherlands
sander.meijer@vka.nl

Abstract. The Dutch governmental Sociale Verzekeringsbank (SVB) recently completed a project on business modeling. The aim was to develop business models that express the strategic goals of customer event orientation, integrated customer services as well as improved agility. An approach was chosen to combine event definition, dynamic process modeling and business rules. In this paper we present our findings. Main conclusion is that, although we used some best practices and common techniques, we had to invent our own ways to combine event definitions, dynamic processes, object models and business rules.

Keywords: Life event, Service Orientation, Business Rule, Dynamic Processes, Governing Process.

1 Introduction

1.1 Situation

The Sociale Verzekeringsbank (SVB) is the organization that implements national insurance schemes in the Netherlands. The SVB takes care of paying child benefit, state pension and other related social insurance payments for Dutch citizens, living inside or outside the Netherlands. The SVB has been doing that for more than a century, by order of the government, and counts some 4.9 million clients.

The organization has about 3500 employees, has implemented about 15 legal regulations, pays about 33 billion euros per year and has a strategy that is directed towards servicing the Dutch citizen excellently. Customer service is characterized by a combination of operational excellence and customer intimacy, and is organized in three main forms:

- Fully automated payment services; the customer receives the benefit without the need to apply for it.
- Self service via internet (www.svb.nl); intelligent dialogues guide the customer through the regulations.
- Service teams, supporting the customer at local desks.

1.2   Challenges

The SVB is continuously working on the improvement of its service. Some main strategic subjects are:

– **Customer event orientation and integrated customer services:** making the service as complete as possible from the perspective of the customer, instead of the perspective of the regulation. When e.g. a customer informs the SVB about a move abroad, the service includes all regulations that are affected by this move.
– **Improving agility:** making IT even more flexible and agile than it already is, in order to be able to adapt the execution of existing regulations easily, or to start executing new regulations within a small time window.

In order to achieve these strategic goals, the SVB has formulated a target-architecture with business, application and infrastructural elements. The architecture contains three main notions:

– **Service orientation:** Service orientation is chosen in order to achieve loosely coupled services on a business and application level, which enables the agility targets.
– **Life events:** Life events are considered to be the unit of work. This notion implements the customer event orientation.
– **Business rules:** Business rules are meant to separate concerns between business logic and technical infrastructure, in order to achieve agility. It is also meant to diminish the complexity of process models, by regulating the process dynamically through rules.

1.3   Business Modeling Project

In 2009 the SVB started a business modeling project with the main purpose to prepare for the implementation of new, rule-based technology, based on the target architecture. The project had three main objectives:

1. The project was meant to model life event orientation.
2. The project was meant to model a generic process that is able to deal with all products and services in an integrated way, in order to support the purpose of integrated customer services.
3. The project was meant to translate legislation to business rules. The scope of the project was limited to a specific business domain: the voluntary insurance for state pension in case of a move abroad[1]. The legislation is relatively complex, and has resulted in more than 250 business rules.

---

[1] Dutch citizens have state pension insurance by law, leading to a state pension from the age of 65. Once a Dutch citizen moves abroad the insurance stops, which leads to a lower state pension at 65. Dutch law offers the possibility of filling this insurance gap by allowing a Dutch citizen living abroad to pay the premium for the insurance voluntarily.

The project results were targeted at a series of models; most notably a List of customer life events (also: life events), a set of business rules, a process model, a business vocabulary, an object model, and other (non-functional) requirements.

1.4   Research Approach and Overview

The article reports how we approached business modeling. There was no a-priori research question, but a result driven mandate to create business models. The findings in this paper are therefore not the result of a structured research methodology. Experience papers like this one are valuable to practitioners and scientists, even if no structured research methodology is followed. This is shown by papers like [2] and [3].

The remaining of the article starts with paragraph 2 explaining the basic approach and concepts, paragraph 3 illustrates some issues that the project had to deal with. The last paragraph 4 contains some conclusions and invitations for research.

2   Basic Approach and Concepts

2.1   Basic Concepts

The business modeling project used the following basic concepts and their inter-relations. They can be considered to be the basic ontology for business modeling. A Customer life event (short: event) is the unit of work for the SVB from the perspective of the customer (citizen), it is where it all starts. Events trigger governing processes. The Governing process (see Paragraph 2.5) is the process that orchestrates other business processes:

- Governing processes orchestrate business processes
- Both governing processes and business processes use business rules

A Business process is a unit of work internally. Business processes use, change or create business objects and use business rules (expressing the decisions and calculations being made in those business processes). A business rule represents desired or mandatory behavior of the organization (see Paragraph 2.3). Business rules restrict or constrain the relations between business objects, which are the concepts and artifacts that are managed by the SVB.

2.2   Life Events

Life events "describe situations of human beings where public services may be required" [6]. Life events focus on a citizen perspective and can be seen as a metaphor for identifying public services related to specific situations that citizens face, rather than the execution of specific (parts of) legislation. Examples of life events are death, emigration, marriage, founding of a company, hire employees. Using life events fitted the goals of the SVB (Paragraph 1.2) with respect to
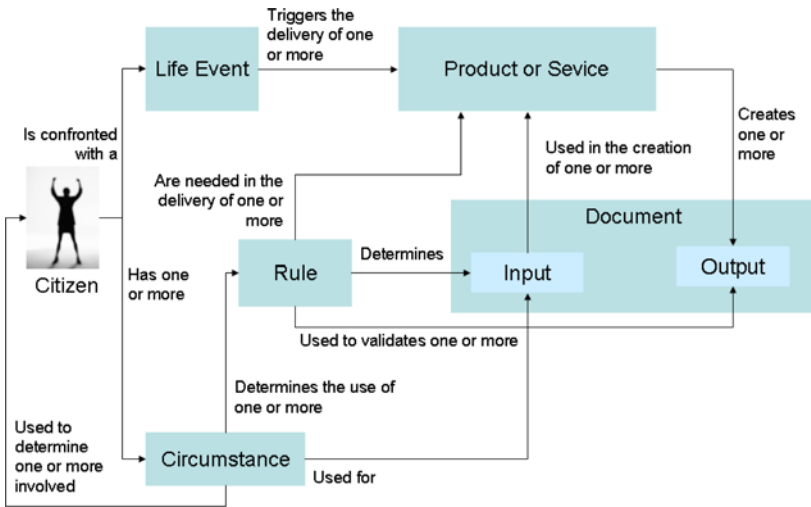
Fig. 1. Ontology guiding modeling effort

integrated customer service. A life event may trigger more than one service. To structure and guide the modeling effort, the ontology outlined in Figure 1 was used, based on [11].

As the figure shows a Citizen can be confronted with one or more life events and one or more circumstances. The circumstances are needed to determine the legislation (coded in rules) that are needed in the delivery of the product/service. Circumstances can also be used for the creation of products or services. Finally the circumstances can be used to determine other citizens, e.g. relatives, that are involved in the same life event. In other words, a life event triggers the delivery of services which are realized by means of processes which rely on business rules. The situations in which a group of citizens, mainly families, is involved (e.g., a whole family moves abroad) is special since it requires service delivery in parallel. We will discuss the consequence of this issue in more detail in Paragraph 2.5. The following is an example of the definition of the life event "moving within the Netherlands":

Example 1 (A life event):

> **life event** -
>   *Moving in the Netherlands*
> **Short description** -
>   *One or more related persons living in the Netherlands, are going to live at another address within the Netherlands*
> **Explanation** -
>   *This life event is used for moving within the Netherlands. It includes the relocation of a child or a roommate. This life event is not applicable for*

temporarily residence elsewhere. Also the change of the correspondence address is excluded from this life event

Circumstances -

(a) There is a new residential address, (b) Current residence = Netherlands, (c)New residence = Netherlands, (d) Residential address = Address = Correspondence Address , (e) Residence time = undetermined, (f) Location != nursing home, prison or other institution

Related products and services [2] -

(1) AOW Pension (2) Child support (3) Anw benefit (4) TOG allowance (5) Voluntary insurance (6) AIO supplement
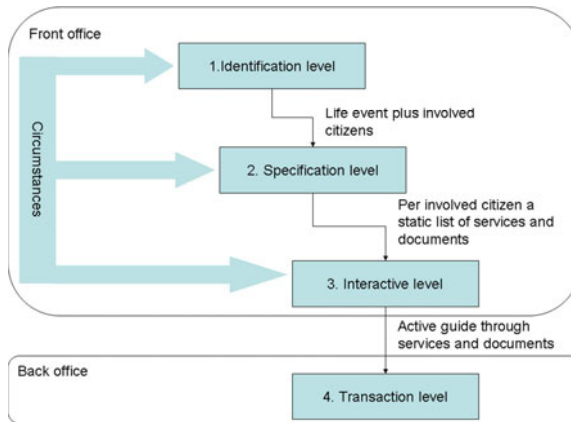
◇

Fig. 2. Life events in relation to governing process

On the basis of this meta model life events can be modeled. When using life events in real situations, there are four different levels of abstraction from the identification of the life event that is relevant to the actual service delivery [10]. We adapted that model to our specific situation and started using this model with four abstraction levels. The adapted model with the four abstraction levels are presented in Figure 2. The relation with the governing process is explained in Paragraph 2.5. The first three levels take place in the front office. In the first level the relevant life event is selected. Since the importance of the circumstances in the identification of all the involved citizens, some circumstances are needed in this first level. After this identification the public products and services can be presented per involved citizen so they can be selected. In order to make this selection possible, additional information is needed about the services like regulation, needed formal documents and timing aspects in the delivery of the service. In the third level the circumstances of the citizen are used to determine the needed input per public services which has been selected in the specification level. In the fourth level the service is delivered through the back office. Since all

[2] The names and abbreviations of the following services refer to Dutch legislation.

circumstances are used in levels one, two and three no circumstances are needed in level four. A governing process is needed to make sure that service delivery lead times from identification until actual delivery are according to law. With these two models we were able to model and work with life events.

2.3   Business Rules

One of the basic principles in the business modeling approach of the SVB is that business rules should not be hidden in process models. As a consequence, rules regarding legislation and decision making are separated from rules governing the flow of processes (See also Paragraph 2.4) (e.g. [8,4]). It is most often the 'know' (the rules regarding legislation and decision making) that must be easily adaptable, whereas the process (the 'flow') can be kept more stable and generic. Therefore it is said that only organisations that manage to quickly adapt and (re)deploy their rules with respect to the 'know' can be truly agile.

One of the conventions in the SVB business modeling approach is to define business rules in natural language with RuleSpeak [8]. Natural language and readibility for business users and legal experts was a prerequisite; any format that would look like programming code or use variables and functions were considered unwanted. RuleSpeak was found to be appropriate, being a set of practical guidelines for declarative natural language statements, based on ORM [5] which on its turn was also the basis for SBVR [9]. RuleSpeak combines well-structuredness with readability. Rules in RuleSpeak are structured according to the ORM and SBVR basic concepts of terms and facts. Terms are collected in a business vocabulary and express the business concepts from a business domain. Facts relate these terms in a meaningful way, can be modeled using a fact model and rules can be seen as constraints on such a model. This is illustrated in the following example.

Example 2 (Illustrating terms, facts, and rules):

Suppose the vocabulary has clearly and unambiguously defined the terms `customer`*,* `age`*, and* `insurance`*. Fact types that may be crafted with these terms are:* `Customer has age` *and* `Customer applies for insurance`*. A rule using these facts could be:* `A customer may only apply for an insurance, if he is older than 18.*   ◇

RuleSpeak gives sentence patterns and rule keywords to express the constraining or deriving rule statements that are built with facts. The rule keywords are *must* and *only*, and every business rule must contain one of these words.

Relevant legislation was thus formalized into sets of rules. We found that legislation can not be translated one-on-one to business rule statements. Instead, we carefully built a domain model and defined the business rules accordingly. Put differently, we have observed that in modeling rules from Dutch law, the structure of the law and the order of the articles is not necessarily mirrored in the (granularity of the) rules. Where the original law text gives criteria for both eligibility and duration in one article, we chose to model the rules on eligibility in a different rule group than the rules on duration. Also, RuleSpeak advises

to express a computation in words, not using the symbols for mathemetical operations. The SVB recorded its own guidelines allowing formulas in business rules, since Dutch law texts incorporate formulas with letters and mathematical operators.

2.4   Dynamic Process

The traditional process modeling approach results in set of fixed patterns, one for each service or product. This fixed way of modeling entails, among other things, that generic activities such as receiving and digitising messages, or sending notifications to customers are modeled multiple times.

A key driver in the new approach focussed on life events is to optimize and reuse processes whenever possible. This has implications at two levels. This implies that the way of working should be decomposed in reusable "chunks" that can be combined dynamically to handle any given life event. The Lego analogy goes a long way in this example: using the different types of Lego blocks one can create structures of practically any kind. Even more, the Lego blocks do not "know" what type of structure they are / can be part of. Indeed, the "chunks" should be defined in such a way that they are (a) independent of each other, and (b) recognisable as specific tasks to employees of the SVB.
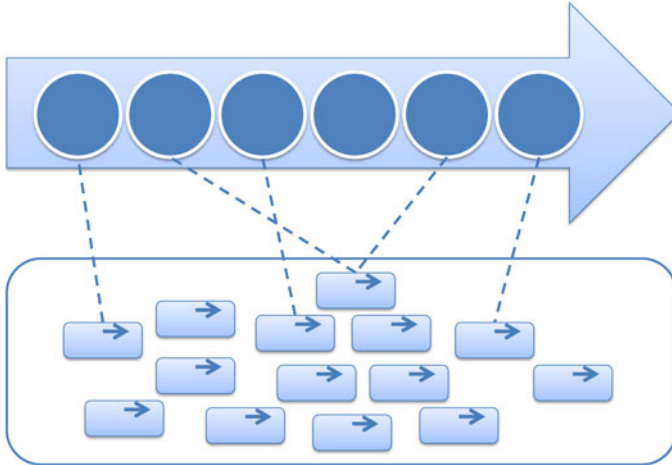

Fig. 3. Handling a life event based on reusable units of work

In the new way of working we have chosen to use a dynamic approach to processes in which generic process steps are combined and reused (more than once, if necessary) to handle the work associated with handling a life event. Figure 3 illustrates this. The figure shows that the process in which a life event is combined by reusing several generic process steps. Even more, one step is used twice.

Our approach to dynamic processes relies heavily on using a combination of process models and business rules. The use of (flow) rules to govern the (order of) execution of processes is explained in Paragraph 2.3 and 3.3.

2.5   Governing Process

Due to the dynamic process modeling approach it is necessary to include process that orchestrates the order in which process steps are executed. We have dubbed this process the "governing process", reflecting the fact that it governs the proper execution of the handling of the life event. The key goal of this process as implemented by the SVB is to ensure that a life event is handled correctly and in a timely manner. The correct handling of a life event implies that assessment of the impact of the life event must be correct in terms of which services are applicable to this particular event. The timely handling of events lies in the fact that legislation determines how long the SVB has to deal with events, taking into account certain exceptions and catering for delays when clients respond slowly/poorly to questions.

This has two consequences (1) as soon as the SVB receives notice of some life event, a governing process for this event must be initiated. This process "lives" as long as it takes to handle the life event. (2) the primary goal of the governing process is to ensure that the right process steps are executed in the right order, while keeping a constant eye on the amount of time that is still available to deal with the event. To keep track of progress, the governing process therefore routinely exchanges status information with the process steps it governs. The complicating factor in designing a governing process that implements these functions lies in the fact that a single life event may have impact on more than one person. Even more, for each person, the life event may have impact on different services (with underlying legislation, encoded in business rules as explained in Paragraph 2.3).

The basic working of the interaction between the operational level and the governing process is illustrated in Figure 4, using the process modeling notation as used in the tool BiZZdesigner [12]. This example shows the different process steps (at the bottom), the governing process (at the top) as well as the interaction points between them. As can be seen, some interaction points are reused
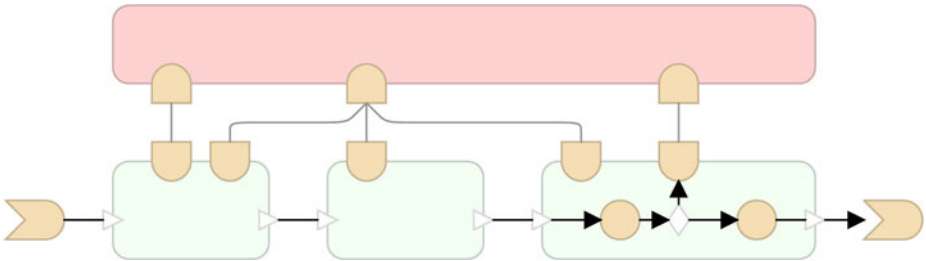

Fig. 4. Governing process

between process steps. This is the case for generic things like exchanging status information. Other interaction points are unique per process step. This is the case for e.g. starting and stopping work, sending information requests to the client et cetera. Note that the interaction mechanism allows for a "plug and play" architecture, where different types of process steps are plugged into the governing mechanism.

The governing process has three levels. All status information from the process steps flows primarily to the service level. This is where we keep track of how long weve been busy with a piece of work; how many days we have remaining etcetera. At the person level, we keep track of different tasks per person. That is, based on information of the service level (note: the service level deals with services with respect to a single person), we attempt to bundle work and interactions with the customer. The person-level reports back to the life event level. This is where we manage the handling of the entire life event for different people. Since most of the work is governed in lower levels. The following example illustrated typical actions in each of the three levels.

level	responsibility & actions
Service level	During the execution of a task it becomes clear that we have to send a letter to the customer to ask for more information about his situation. In this case, the timer that keeps track of how much time is left for dealing with this task has to be stopped until the answer is in.
Person level	For two different tasks (two different services) we need information about the current income of the customer. We only want to ask the customer for this information once. Therefore, the person layer keeps track of what weve asked the customer and when.
Life event level	When new information arrives about a current case (e.g., the answer to a written question), this level ensures that the new information is coupled to the ongoing handling proces(ses) as well as triggers lower-level handling processes that timers should be restarted, et cetera.

3   Issues

3.1   Issues with Life Events

One of the main issues that we have identified in dealing with life events lies in the assumption of objective observation. We assumed that it would be possible to objectively identify life events which was not the case, mostly due to the fact that the criteria for distinguishing one life event from another were subjective. Other issues that we have encountered are:

– It is not always clear to whom products and services should be delivered. Due to complex legislation it is possible that a life event affects more than one person, depending on highly specific circumstances. Whether a set of circumstances matches the set of criteria is - in some cases - a matter of opinion.

- The originally identified life event can be wrong after close examination.
  This issue is best illustrated with an example. A (single) person moving, need not always pertain to the life event "moving within the Netherlands", even though all the criteria for this event are matched. It may be that the *actual* life event is a divorce.
- The integrated service delivery is not always determinable in advance.
  Even though service delivery was designed to be integrated as much as possible, it turned out that services may have different lead times. Due to legislation that asserts the maximum time available for handling a life event, it may therefore be the case that outcomes of all services cannot be packaged and communicated in one go.

3.2   Issues with Dynamic Modeling

The core issue that we faced with dynamic modeling follows directly from the lack of a standard notation/ approach to such a way of modeling. More precisely put, current modeling approaches assume that processes are modeled as a consistent whole and *therefore* lack notation to explicitly represent the fact that (the model of) a process can be considered a building block that is used in assembling a larger process similar to the fact that lego's can be used to build a variety of structures.

Consider the handling of a life event whereby the SVB and its customers "take turns" in doing work. This is particularly the case when the information pertaining to the life event is not all that clear, or when information is missing. An example of was given in Paragraph 3.1 when we discussed that moving within the Netherlands may actually pertain to the life event "divorce". Asking for more information has the effect that it is "the customers turn" to get some work done which may take a while. As new messages pertaining the same life event reach the SVB, generic process steps may have to be executed more than once in order to succesfully handle a life event.

This brief example shows that, due to the dynamic approach, process models are no longer linear. Since we had slightly amended standard modeling language, and re-interpreted particular concepts in these languages, the models *appeared* (visually) to be linear, especially since they closely resembled the traditional process models that are used in the SVB in terms of notation and presentation. Indeed, we found that stakeholders read/ interpreted these models as being linear. As a concsequence, more time than expected had to be invested in explaining our models to different groups of stakeholders.

3.3   Issues with the Cohesion of Models

A dynamic approach to modeling life events depends on the cohesion between several models (business rules, process model, underlying fact model). Communicating each of these *individual* models with respective groups of stakeholders is challenging, as is a well-known fact in any modeling approach. However, these communication issues are compounded significantly when communicating the

the *combined* set of models cohesively to different stakeholders such as (a) the business, i.e. the people who will actually perform the work in a new way, (b) management, having to decide upon the details, (c) the legal department, checking if the complexities of the law have been implemented properly, (d) the IT services department.

Since the SVB has achieved excellent results with a process atlas in the past, it seemed natural to take process models as a starting point; the assumption being that the type of process models that have been used so far are widely understood, yet formal enough to function as specifications for a software system as well. Even more, from previous research [13] we knew that approaches based on process models are easily combined with business rules. Indeed, business rules have been used extensively in order to govern execution of process models, to codify legislation etcetera. This did leave us with several challenges pertaining to process models, rules, and their combination:

– many languages are available for modeling business rules, varying in the level of formality and readability by non-technical stakeholders. In a previous pilot project, SVB had experimented with operational rules (Event-Condition-Action), which seems to fit best in event-driven situations. The readability of this form for legal experts, however, turned out to be a problem. The paradoxical situation arises that rules should be human-readable for business users, while at the same time be formal and specific in order for a machine to interpret and excute them. It was therefore decided that a two-pronged approach was to be used for the time being: specify the rules in natural language (but as formally / structured as possible) and convert them to an execution platform later. This allowed for easier / better validation by the legal department and allows for a less steep learning curve. At a later point, full-blown SBVR might be introduced to let the two steps converge into one [9].

– In combining process models and rules, we had several challenges to overcome in terms of tooling (we used different software tools for modeling processes and rules) and technique. As explained in Paragraph 2.3, we distinguished between 'know' and 'flow' type of rules, each of which may have to be excuted in a single activity (of a process step). A distinction that is made, then, is the logical excecution of rules on the one hand, and the application service that may have to be called to gather the required information to be able to execute them.

This is illustrated in Figure 5: before we execute an activity, we firstly execute the flow-rules associated with this activity. This determines, among other things, whether the activity should be executed at all. When the activity should indeed be executed, the data that are necessary for executing know-rules are retrieved using application services. The rule set also determines if additional data is missing. If this is the case, a dialogue may be necessary so that the missing data may be added.
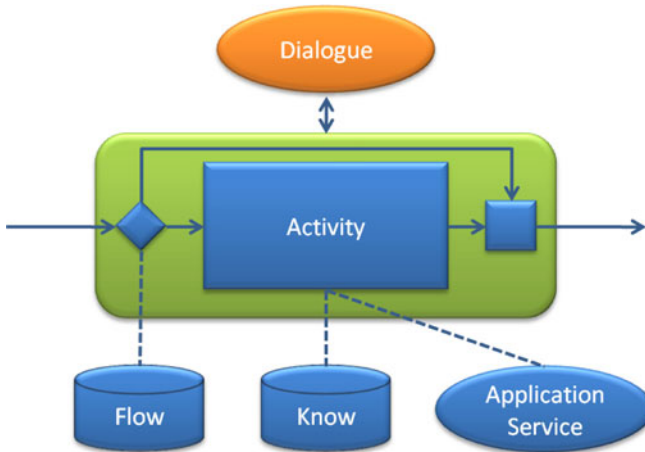
Fig. 5. Combing rules and processes

3.4   Project Organizational Issues

As described above, the total set of business models consisted of process models, business rules and data models. In a project of this size it is a challenge to organize a project team in such a way that all aspects of business modeling are represented, and more importantly that the models are integrated and coordinated.

This issue was addressed by forming a multidisciplinary team consisting of several subteams, including a business rules team, a process modeling team and an object modeling team. The teams were led by two teammanagers who were responsible for planning and the day-to-day management of the subteams. The team managers reported to one project manager who was responsible for the progress of the overall project and who managed all external stakeholders. In addition to the aspect teams, an experienced enterprise architect was involved as project architect. His role was to counsel the teams methodologically, and who to guard compliance to the referential enterprise architecture. The project architect also managed the communication from the teams back to the counsel of enterprise architects.

The project team worked in a strong iterative and incremental fashion. The work was structured into work packages of four to six weeks, depending on the relative size of a topic. This ensured that the different model types were coordinated for a formal delivery at least every four weeks. An major challenge was to manage the dependencies between the teams during these work packages. In order to facilitate close interaction between the teams we held near-daily stand-up meetings. These stand-up meetings, which were held in the morning at the start of a work day, were intended to last fifteen minutes. During these sessions each subteam talked about what tasks they have been working on since the last session, what problems they encountered and whether there were dependencies

between teams that need to be addressed. Each work package followed the a Plan-Do-Check-Act quality cycle [1]:

- **Plan:** In a workshop with the different aspect teams, the high level tasks were broken down to a task level. The duration of the tasks were estimated and the tasks were assigned to teammembers.
- **Do:** The actual work during a workpackage was performed as indicated in the above paragraphs.
- **Check:** At the end of a workpackage each aspect team did a review session. During this review session we evaluated the workpackage and discussed what practices to retain and what practices to discard.
- **Act:** the suggested corrections and improvements in the process were implemented into the work process. The new principles and renewed process was documented in a project wiki.

In this way we followed a continuous improvement cycle necessary for a team that was working in an innovative environment. Although the team composition, the iterative working style and the frequent meeting, an important challenge remained. It was very hard to find a good balance between meeting and coordinating with each other, and to actually get any work done. Especially in the first few months of the project, the stand-up sessions took more than the planned fifteen minutes.

4   Conclusion

Business modeling has been a new discipline for SVB. We both adopted several best practices and developed our own methods. For event definition we used some best practices but had to solve ourselves some complicated issues. For business rule definition we used SBVR related RuleSpeak and enriched it with some specific patterns. For object modeling we mixed UML and ERD, and had to invent its relationship with rules, based on ORM. For dynamic process modeling we quickly learned that existing languages like BPMN all assume that processes are modelled from start to end. In order to introduce the dynamic aspect in our process models - i.e., using generic building blocks - we had to adapt existing approaches in terms of approach and notation.

Based on these experiences we can conclude this article with a couple of recommendations, as well as with several subjects that need further research because of the lack of theories and best practices.

4.1   Recommendations

- Use an iterative way of working with a multidisplinary team. Communication was quintessential to our project, with different stakeholders as well as within the modeling team. We found it to be a challenge to organize a project team in such a way that all aspects of business modeling are represented, and more importantly that the models are integrated and coordinated. Only by

working in a strong iterative and incremental fashion did we manage to align the efforts of different modeling-groups, and keep models consistent and focussed.

– The use of RuleSpeak resulted in a very smooth validation process with business users and legal experts, due to the natural language character of RuleSpeak and the lack of technical jargon.
– The use of proven (meta-)ontologies like the one for life events gave us a jump ahead and is a recommended practice.

4.2   Suggestions for Further Research

– There is a need for a new notation method for dynamic process modeling. The main requisite is to be able to present the process models in an understandable way for business users without losing the dynamic character.
– There is a need for a methodology that incoporates the cohesion between rules, processes and objects (or concepts). SBVR and ORM provide a formal relationship between rules and concepts (terms), but a convention for modeling the relationship between rules and process models is lacking.
– Defining life events needs more sophistication in order to enable modeling the detailed cases we were encoutering. The intuitions that were needed to assess the right life event, as mentioned in Paragraph 3.1, can be read as an invitation to analyse these intuitions and to create a life event domain knowledge model.

All in all, we experienced that using a dynamic (process) modeling approach with business rules in order to implement life events can be succesful.

References

1. Dwing, W.E.: Out of the Crisis. MIT Center for Advanced Engineering Study (1989)
2. Richardson, G. L. et al.: A Principles-Based Enterprise Architecture: Lessons from Texaco and Star Enterprise. In: MIS Quarterly, pp 385–403 (1990)
3. Stader, J., Bridge, S.: Results of the Enterprise Project. In: Proceedings of Expert Systems 1996. In: The 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems, pp. 888–893 (1996)
4. Burlton, R.T.: Business Process Management: Profiting from Process. Sams Publishing, USA (2001)
5. Halpin, T.: Information Modeling and Relational Databases - From conceptual analysis to logical design. Morgan Kaufman, Academic Press (2001)
6. Wimmer, M. and E. Tambouris: Online One-Stop Government - A working framework and requirements. In: Information Systems, pp 117-130 (2002)
7. Business Rules Group: Business Rules Manifesto - the principle of rule independence. Version 2.0 (November 2003)
8. Ross, R.G.: Principles of the Business Rule Approach. Addison-Wesley, Reading (2003)

9. Object Management Group: Semantics of Business Vocabulary and Business Rules (SBVR). Technical specification no. dtc/06-08-05 (2006)
10. Todorovski, L., Leben, A., Kunstelj, M., Cukjati, D., Vintar, M.: Methodology for Building Models of Life Events for Active Portals. In: Wimmer, M.A., et al. (eds.) Proceedings of 5th International Conference on Communication. Springer, Heidelberg (2006)
11. Trochidis, I., Tambouris, E., Tarabanis, K.: An Ontology for Modeling Life-Events. In: IEEE International Conference on Services Computing, pp. 719–720 (2007)
12. BiZZdesign:Handboek Business Process Engineering (in Dutch). Van Haren Publising (2008)
13. Gong,Y., Janssen, M., Overbeek, S.: Zuurmond: Enabling flexible processes by ECA orchestration architecture. In: Proceedings of the 3rd International Conference on Theory and Practice of Electronic Governance, pp. 19-26 (2009)