



Why are Business Process Models often too complex? Do's and Don'ts for Business Process Modelers

Version 1.0

This document developed by

Dr. Juergen Pitschke, BCS-Dr. Juergen Pitschke, www.enterprise-design.eu

This material may be used freely on an in-house, non-commercial basis. Commercial re-use or re-transmission of any portion of this material is prohibited without written permission of BCS – Dr. Juergen Pitschke. Contact BCS – Dr. Juergen Pitschke for licensing and re-use arrangements. Please include this notice in all reproduction.

Table of Content

1	The Question	3
2	Typical Problems and Solutions	3
2.1	Avoid Complex Sequence Flows – Reuse Model Elements	3
2.2	Use Different Levels of Abstraction – Decomposition versus Composition.....	5
2.3	Use a Standard Notation	6
2.4	Understand the Model Elements you use	8
2.5	Don't Expect to Show All Aspects of a Business Process within a Visual Model	10
2.6	Be Consistent with Terms in Naming and Documentation	10
2.7	Focus on the Business Perspective.....	11
2.8	Separate Independent Concepts.....	12
2.9	Support Traceability	12
2.10	Use Tools which are easy to use.....	13
	Literature	14

1 The Question

Ron Ross, the father of Business Rules, asked in November 2009 on LinkedIn: "Why Are Business Process Models Often So Complex? Do you agree?" The first reaction of many business process modelers to this question is "Are they?" If we look honestly on process models, the answer is yes. Many process models are real art work, hard to understand by other people, hard to keep up-to-date. Often customers ask me for help with such problems to make the models usable, maintainable and this way valuable.

If you ask me for reasons why models may be too complex some general issues come to my mind immediately:

- People often start to model intuitively. There are no guidelines, no naming conventions, etc. When the project grows, models become inconsistent and are not maintainable.
- Very often the modeling project starts on very fine detail. Processes are described on the task level. This usually ends up in the big, unstructured wallpaper. A structure is missing. Nobody can use the model.
- Modelers and readers misunderstand elements of the used notation or don't use a standard notation at all.
- Modelers mix up different perspectives in one model. It seems to be hard for many people to really separate the business view from implementation details.
- Modeling projects start often in the mood "Everybody can do this." No planning, no systematic approach, no working techniques.

To avoid complex, unusable process models you need to handle modeling as an engineering discipline with defined working techniques and methods.

2 Typical Problems and Solutions

Let's have a look on some typical issues and possible solutions. Don't expect a panacea but some insights to improve your own models. This article is not intended to be a systematic approach to business process modeling.

2.1 Avoid Complex Sequence Flows – Reuse Model Elements

Look on figure 1. Can you say what the model shows? It should not take longer than a minute because it is showing something very simple.

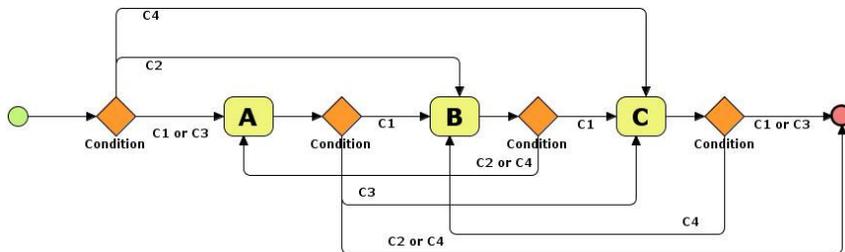
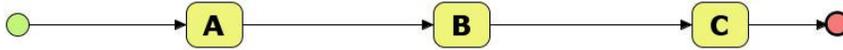


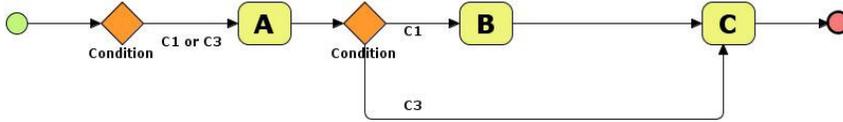
Figure 1: Sample BPMN Model

Don't laugh at the example. I see such diagrams much too often. Review your own models and I guess you will find such cases too. On the other hand bad examples are much more entertaining.

How was this model created? First the modeler created a model for the happy-day-scenario.



Then he started to add exceptions and additional scenarios.



Finally he did end up in the mess.

Can the model also be shown in another way, easier to understand and easier to maintain? Yes. Figure 2 and 3 show two different ways to show exactly the same content in different ways.

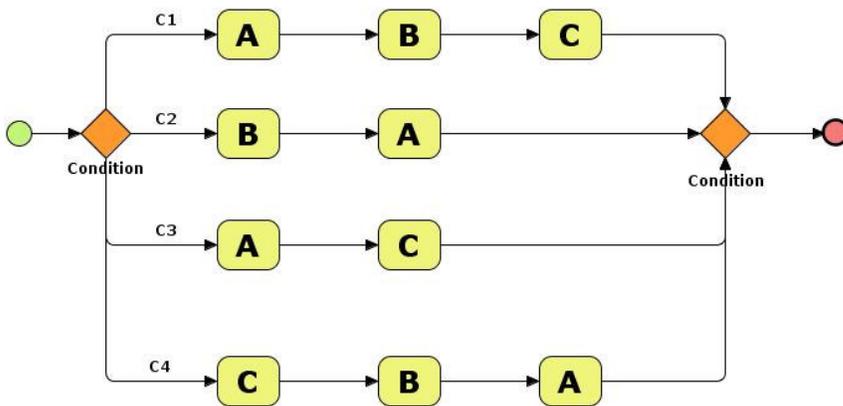


Figure 2: Diagram 1 redone – option 1

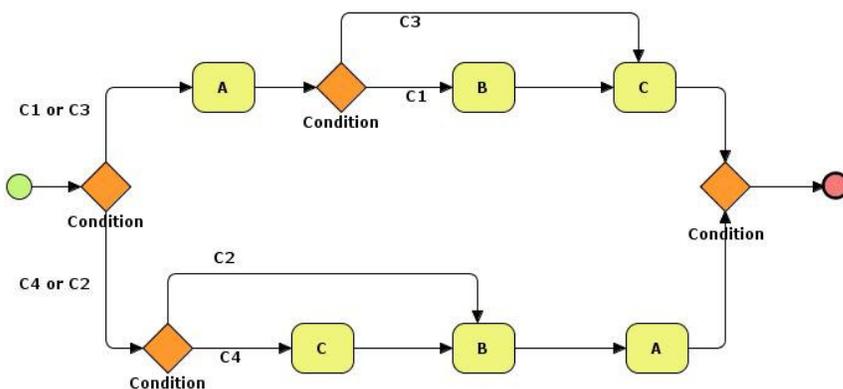


Figure 3: Diagram 1 redone – option 2

As said, both diagrams show the same content as Figure 1. But both diagrams are much easier to understand and to maintain.

Are both versions equal? With regard to the content shown in general, yes. Semantically I don't think so.

If somebody reads figure 2 he/she would see four different scenarios to perform the activities A, B, and C which are equally important and equally likely.

Figure 3 tells us that there are two different scenarios, both equally likely and important. In the first scenario the activities A, B and C are performed in this sequence. In the second scenario the sequence is reversed and the activities are performed in the order C, B, A. Both scenarios have one exception. In the first case you can skip B; in the second scenario you can skip C. To decide which is closer to what we want to show we need to know the business content instead of talking about abstract activities.

To make the diagram readable we reused some model elements and did show them more than once in the same model. Many modelers hesitate to reuse model elements. A reason is often the tool used. To be able to reuse model elements you need a tool which is repository based. If the modeling tool uses a repository it is possible to use the elements multiple times in different scenarios or views, but to describe each element only once.

Modelers sometimes argument that reuse of model elements makes the model "bigger". As you can already see in our small example this is not true at all. We have a higher number of activities in our model, but the total number of model elements is reduced.

There are more possible solutions to fix our original, messed up model. E.g. show the different scenarios independently, describing the context of each scenario. But in each case reuse of model elements is the key to improve the model.

To find a solution as in figure 2 or 3 you need to think first on a higher abstraction level. Think about the *business content* first: Which business scenarios exist? Which are relevant? Don't think about conditions and condition C1, C2, C3, C4 too early.

Do: Reuse model elements to avoid complex sequence flows.

Do: Think business content instead of implementation level details.

Do: Use a modeling tool supporting reuse of model elements.

2.2 Use Different Levels of Abstraction – Decomposition versus Composition

Part of the problem we saw in 2.1 is that many modeling projects start on a very fine level of detail. Business processes are shown only on the task level; tasks have a very high granularity. Again: Often Tasks are more on the implementation level of detail instead of the business level. In the result the process models are poorly structured, the models are very big and hard to manage.

To avoid this we want to show business processes on different detail level – starting with a general structure to the details of the task level. We don't want to show processes on the task level only.

How many levels do we need? This depends ...

The magic number seems to be three. To drill down more than three levels is not really feasible in daily practice. Exceptions are just exceptions and don't prove this rule.

In our modeling projects we define three levels of detail for the business processes:

1. Macro Level
2. Sub-Process Level
3. Task Level

The modeling guidelines contain rules and specifications for each detail level.

- Macro Level Processes show only the general structure of the process. A Macro Level Process contains only sub-processes. Only the "Happy-Day"-Scenario is shown. Exceptions are not of importance on this level of detail. The focus is on the sequence of the sub-processes, information exchange is not shown.
A Macro Level Process contains between 5 and 10 sub-processes.
- Each sub-process shown on the Macro Process Level is refined on the Sub-Process Level. A model on the Sub-Process Level shows the structure of the sub-process in greater detail as the Macro-Process Level but doesn't go down to the Task Level. The focus is still on the structure of the sub-process. We show the happy-day scenario, alternative scenarios and important exceptions. Information exchange between process participants is shown including used information objects.
A model on the Sub-process Level contains approx. 25 activities. The focus is still on sub-processes but it is possible to show some tasks too. If a sub-process contains much more than 25 activities review the Macro-Process Level and try to find a better structure.
- Each sub-process on the Sub-Process Level is refined on the Task Level. A model on the Task Level shows the structure and the details of the sub-process in detail. We show all *relevant* scenarios and exceptions. We also show the information exchange between the process participants in detail including information objects.
A model on the Task Level contains approx. 25 tasks. Sub-processes are not allowed.

In my experience three levels of detail are just fine. Sometimes customers come up with an additional level of detail. In all cases so far the additional level of detail is already on the implementation level and not related to the *business view* on the business process. This depends also on the understanding of the concept "Business Task".

How can we create a complex model showing the three levels of detail?

There are two general approaches:

- Decomposition:
Start on the Macro Level and decompose the sub-processes to the Sub Process Level and then to the Task Level.
- Composition:
Start on the Task Level and construct the other detail levels from groups of related activities.

A good modeling tool should support both working techniques: decomposition and composition.

The more common and easier way to construct a model is decomposition. However sometimes a model on the task level is given and we have to start with this. Starting on the Macro Level helps us to think about the business scenarios first as discussed in paragraph 2.1.

Most modeling processes today are iterative. This means both techniques are applied within a project. To start with decomposition is the preferred way in our own approach.

Do: Show your Business Process on different levels of abstraction. A three-level approach is suggested.

Do: Define the granularity and cardinality for each detail level.

Don't: Don't show the process on the task level only.

Don't: Don't start on the task level.

2.3 Use a Standard Notation

To describe business processes you need a notation rich enough to show the relevant aspects of a business process.

In the literature you find different notations, depending on the preference of the author and the use of the process model:

- Informal Swimlane Models
- BPMN (Business Process Model and Notation) based models
- EPC (Event-Driven Process Chain) based models
- Activity Diagram (part of the UML – Unified Modeling Language) based models

Swimlane Diagram

Swimlane Diagrams are used since a very long time. The concepts are familiar to many business users. However I see two major shortcomings:

- The elements are not formally defined. Misunderstanding and miscommunication is programmed especially when using such models for a long period of time.
- No differentiation between control flow and information flow is made.

Because BPMN Diagrams incorporate the elements of swimlane diagrams most modelers switch to BPMN based diagrams these days. If the BPMN elements are used thoughtfully business users pick up BPMN diagrams as easy as swimlane diagrams.

BPMN – Business Process Model and Notation

BPMN is the OMG standard to describe, implement and execute business processes. It is a rich notation, covering different views on a process.

Critique is coming from different directions:

- The notation is covering modeling AND implementation of business processes. The new version of the standard (BPMN 2.0) is very focused on the implementation level.
- This causes that the standard becomes very complex and for a “normal” business user difficult to adopt.

There is a good message: Nobody forces you to use all elements of the standards. In daily practice you need to define a subset of the BPMN (“Essential BPMN for Business Modelers”) you want to use to describe business processes on the business level. In my experience business users adopt such a subset of the BPMN fast and are able to use BPMN based process models.

If you need help in defining the subset which is suitable for your project, please contact me.

A big benefit of BPMN is the fact that it is an accepted standard. We see a high adoption rate in different industries. Students leaving the university are familiar with the basics of the notation. More and more people use the standard in their work. This is especially important when you create process models which should be used and maintained for a longer period.

A benefit compared to the informal swimlane models is:

- The notation allows support for decomposition and composition of a business process
- Clear differentiation between sequence flow and information flow
- Separation of concerns: Different views and model elements for describing process flow and communication between process participants
- The standard defines an exchange format between tools supporting the standard

EPC – Event Driven Process Chain

EPCs are originated in the ARIS toolset and most often seen in the context of SAP implementation. Compared to informal swimlane models and also to BPMN models EPC Diagram are much more formalized. The notation is close to implementation. You find e.g. implementation level details as “entities”, “databases”, etc. Most EPC tools support BPMN too now.

Activity Diagram (UML – Unified Modeling Language)

UML is the notation for design and development of object-oriented software systems.

Proponents position Activity Diagrams as a possible way to describe business processes. I

don't suggest this way for business users.

There are similarities between Activity Diagrams and Swimlane and BPMN Diagrams with regard to basic constructs as the use of swimlanes (named Activity Partition in UML), actions and activities. However if you look into the details of the Activity Diagram object-oriented programming is grinning at you. For a business user it is difficult to use the concepts to describe business processes. Some needed elements are missing. E.g. the Activity Diagram does not include an "Inclusive OR" Gateway as the BPMN. This is understandable from a programming standpoint. The programming languages don't have a construct for an "Inclusive OR". For a business modeler the optional gateway is needed to show optional scenarios of a process.

The named notations focus on the description of process flow and communication between process participants. There are more notations and techniques available to describe other aspects of a business process. E.g. with IGOE you describe the details of a business activity. If this is relevant to your project such concepts can easily combined with BPMN or other notations.

Even if BPMN is not perfect a subset of BPMN is the best choice available today to visualize Business Process Models. BPMN is an accepted standard in many industries.

Do: Select a standard notation to describe your business processes. The notation should cover the relevant aspects of a business process and should be familiar to the modeler and the reader.

Do: Select a subset of the chosen notation for your project.

Do: A BPMN subset is a good choice for describing business processes on the business level.

Don't: Don't use a notation which is focused on implementation only.

2.4 Understand the Model Elements you use

BPMN is a standard targeting very different audiences. That's why many elements are defined in a way which let a certain degree of freedom for the modeler. You need to reduce this degree of freedom for your modeling team by further detailing the standards and defining modeling guidelines for your team.

A good example is the description of the model element "Lane". The specification says "A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities within a Pool. ***The meaning of the Lanes is up to the modeler.***" The standard specification of BPMN 1.2 gave some examples in addition: "Lanes are often used for such things as internal roles (e.g., Manager, Associate), systems (e.g., an enterprise application), an internal department (e.g., shipping, finance), etc. ". If you don't specify what a Lane should be used for in your project you will end up in a mess again.

To decide how you want to use a model element as "Lane" you need to think about content and quality criteria of your business model. Using a lane to show a system is not appropriate on the business level. Using a lane to show a department would make your process model instable. Every time the enterprise changes the organizational structure you would need to change your process model – even if the process itself didn't change at all. The best choice on the business level is to use lanes to present a role. This leads to the discussion of what a role is in this context. Let this topic out for another paper.

There are some elements in the BPMN having a clear definition but are frequently misunderstood even by experienced modelers.

The most common misunderstandings are:

- Sequence Flow versus Control Flow
- Using Gateways

The relation connecting two business activities in a business process diagram is called "Sequence Flow". The specification says: "A Sequence Flow is used to show the order of Flow Elements in a Process or choreography ". The "old" version 1.2 of the standard was more informative and details this: „BPMN does not use the term "Control Flow" when referring to the lines represented by Sequence Flow or Message Flow. The start of an activity is "controlled" not only by Sequence Flow (the order of activities), but also by Message Flow (a message arriving), as well as other process factors, such as scheduled resources. ".

Most modelers and readers of BPMN models often set "Sequence Flow == Control Flow". This is completely wrong and causes bad process models and inefficient process implementations. Version 1.2 of the BPMN standards explicitly points to the fact that the sequence flow is just one of many conditions establishing the control flow. The sequence flow is maybe the less important factor for defining a good control flow, especially in service processes. Focusing on other factors needed for the control flow as needed information, needed resources, etc. can improve the efficiency of a process in large scale.

Business rules can help to specify the control flow in a very elegant and flexible way. It seems to me that this is a highly overlooked aspect of integrating business rules and business processes.

Another misunderstood element in BPMN is the Gateway. The definition says: "Gateways are used to control how the Process flows (how Tokens flow) through Sequence Flow as they converge and diverge within a Process. If the flow does not need to be controlled, then a Gateway is not needed. ... *Gateways do not represent 'work' being done and they are considered to have zero effect on the operational measures of the Process being executed (cost, time, etc.).*"

Especially the second part "... Gateways do not represent `work` being done ..." is often ignored. A gateway is an artificial construct showing alternative, parallel or optional scenarios within a process. This is usually clear for a parallel gateway: We want to show that some activities are performed in parallel. This often becomes unclear when using a XOR gateway to show alternative scenarios. The gateway is e.g. named „Calculate Order Volume". The outgoing paths are named „Order Volume < 1.000 Euro", „Order Volume > 1.000 Euro and <5.000 Euro" and "Order Volume > 5.000 Euro", etc. The name of the gateway "Calculate Order Volume" indicates that this is not a gateway but an activity. Work is performed, we calculate something. The correct solution is to show an activity for the calculation and having the gateway determining which business scenario is executed depending on the result of the calculation.

The other issue with gateways is related to the discussion of sequence versus control flow. Gateways are often used to "control" the flow instead of showing different scenarios within a process. If we really stay on the business level gateways should be used carefully to show different scenarios within a process. If the process structure becomes too complex it is better to show the different scenarios in different models.

The number of gateways in a process model is a good quality criterion. If you use too many gateways you are maybe on the implementation level or you use the gateway in a non-intended way or your model is not well-structured.

Do: Precise the use of the elements you decided to use in your models.

Do: Make the meaning of the elements on the business level clear to yourself and your customer.

Do: Define quality criteria for your project. Apply these criteria honestly.

Do: Help the addressee of your model to understand the model elements in the same way you do.

2.5 Don't Expect to Show All Aspects of a Business Process within a Visual Model

In the first phase of adopting a visual modeling approach many users focus on the diagrams. After some time they realize that the diagrams alone are not enough. The documentation and some properties of the model elements are filled.

In training sessions users often ask "Can I visualize that the documentation is filled?", "Can I see in the diagram that the element has a certain property filled?", etc. In the most cases the answer is no. The expectation that you can see all aspects and properties and details in a visual model is just wrong.

Trying to visualize too many aspects and properties in a single model leads again to complicated, unusable models.

An important statement in our daily work is "Models are much more than pictures!" Each element in a model has a documentation giving additional information in textual form. The notation defines a set of properties for each model element. Both are not visible in the diagram but contain important information.

Which properties you use, and what information you provide in the documentation, depends on the intended use of your model. We see a rich spectrum in our daily work:

- Process models as a base for communication, for work orders and work manuals
- Process models as a part of a request for proposal or as a blueprint for an IT implementation project
- Process models as a tool to analyze, design and optimize a business process
- Process models as a tool to simulate business processes

Depending on the purpose of your model you have to define which elements and which properties to use and to specify the form of the documentation.

When you use the model for analysis you maybe combine the model with approaches as IDEF0 or IGOE. You maybe have to record KPIs. If you want to do a simulation you need to provide information about available resources, needed time for an activity and much more.

The style guide we suggested in paragraph 2.4 will be extended specifying the requested information for each model element.

Do: Specify which properties have to be filled for each model element.

Do: Specify which content should be contained in the textual documentation of each model element. Have a textual documentation for business activities.

Do: Include other proven approaches in your models.

Don't: Don't try to visualize every aspect of the Business Process.

2.6 Be Consistent with Terms in Naming and Documentation

Visual Models are constructed from *VISUAL* elements. The main purpose of a visual model is to support communication. But – communication between people is usually based on natural language, not on pictures. Communication between business people should be clear and without misunderstanding.

If your models need to support communication you need to have a consistent vocabulary you use in naming and documenting model elements. The term "vocabulary" comes from another OMG standard: Semantics of Business Vocabulary and Rules (SBVR). This is not a standard defining a visual modeling notation. So it is by far not as popular as the graphical notations.

This is another common but unfortunate misconception about visual modeling. Without a defined vocabulary your visual model is unusable.

The style guide has to include a naming convention. Independently which naming convention you choose: Make sure the naming convention is used consistently through all models in your project.

The common naming convention for a business activity in a business process is a <<verb>> <<noun>> combination. This is questionable. But let's go with it for the moment.

If you follow this naming convention all <<nouns>> should be defined as terms in the vocabulary. If your modeling tool supports a vocabulary, it is maybe able to check this for you. If a term used in naming an activity is not known in the vocabulary you need to define it to avoid ambiguity.

What's about the <<verb>>? If you follow the SBVR approach the <<verb>> is related to a Fact Type. And here the common naming convention becomes questionable. If we follow a <<verb>> <<noun>> convention we only support unary fact types.

Let's look for an example. We know two terms: Shipment, Container. We also know one fact type: "Shipment is loaded into Container". Following the common naming convention we would name the business activity "Load Shipment" or "Load Container". If we compare this to the fact type it is incomplete. The correct name of the activity should be "Load shipment into Container". To check this name against a vocabulary requires more intelligent tools. It is not enough to check if the terms are defined. We also need to check if the verb is defined as a fact type and if the terms used are included in the fact type.

The sample is just one case and not the most difficult. Let's assume another set of terms and fact types. We know two terms: Help Desk Engineer, Incident. We also know one fact type: "Help Desk Engineer resolves Incident". In this case we name the activity "resolve Incident" and place the activity within the lane "Help Desk Engineer". The naming of the activity and the dependency of the model elements (activity is a child of lane) would reflect the fact type correctly.

We change the naming convention for business activities to "The name of the activity has to reflect the underlying fact type." For most (all?) existing tools it will be difficult to check all possible cases for this. Some modelers will have a problem to understand this naming convention. We have to explain and detail it.

Do: Define a vocabulary as a base for naming model elements in the business process model.

Do: Define an appropriate naming convention for all elements in the business process model.

Do: Use the naming convention consistently through all your models.

2.7 Focus on the Business Perspective

Discussing the use of model elements I already pointed to the fact that we have to focus on the *business* perspective in a *business* process model. This is also true for the content of the model.

Analyzing customer projects I often see implementation details in elements used, in naming elements, in describing elements. Usually we don't start a project on the green grass. There is a process, there are existing IT systems to support the as-is process. In many cases the existing implementations make their way into the daily communication. Sometimes (often) we talk about system functions instead of the ***business*** content of the activity.

One of my customers used the event "Status 13" in their process models often. Nobody in the company complained about the name. Everybody was very well aware what the meaning of "Status 13" is. Except me. I asked about the meaning of "Status 13" and I learned that the meaning is "Shipment left port of departure". OK. Again – don't laugh at it. Each and every company I was working with so far has its own "Status 13".

The existing system marked the event "Shipment left the port of departure." with a status value "13". "Status 13" is much shorter than "Shipment left the port of departure". So after some time the "Status 13" made the way into daily communication within the company. It is not easy to change and fight this. But to make the model usable in mid to long term it is necessary to name and describe model elements with the real business meaning.

Another example for thinking in implementation details is the presentation of communication between process participants. I often see events as "Email Received" or activities as "Send Email" in a *business* process model. I always ask "What is the content of the email?", "Can the content also be received in other ways as fax, internet, etc.?", "Can the email also contain other content?" This way you will find out that the real name and the real *business meaning* of the event is. The event is maybe better named "Order received". The real meaning of the activity is e.g. "Send Order Confirmation".

Do: Specify the business content of your business processes.

Do: Review your models frequently for implementation details.

Don't: Don't use implementation details of existing systems for naming and describing business events or business activities.

2.8 Separate Independent Concepts

In 2.4 we discussed the use of the model element "Lane". My suggestion was to use "Lane" to represent a Role and not a Department within the organizational structure. The argument for this was that the organization structure is instable compared to the process and that the organizational structure is independent from the process and vice versa. Using the Lane element to represent a role and mapping this role to the organizational structure makes our process model easier to use and more stable against changes.

The principle "Separation of Concerns" is also important with regard to other aspects of a business process model.

- Separate Organizational Structure and Process Organization
- Separate Business Process and Business Rules
- Separate Business Requirements from Business Process Realization
- Separate Control Flow within the process and Communication Flow between process participants.

If you try to show different concepts within one model the model will become big, but it will especially become hard to maintain.

Do: Separate independent concepts as Business Rules and Business Process or as Organizational Structure and Process Organization.

2.9 Support Traceability

If you follow the suggestion in 2.2 and 2.8 you need to be able to trace the relations between models and between model elements.

If you decompose a process from the macro level to the task level you always need to know which detail model belongs to which element in the higher level process.

When you separate different concepts, you need to be able to show the mapping e.g. from a Role, represented by a lane, and an organizational unit in the organizational map. You want to document which business rules are used within a business process or business activity. You need to show the connection between a business requirement and the realization of the requirement in the business process.

This is also a tool question. A good modeling tool should be able to trace such relationships. The minimal requirement is to be able to trace decomposition and composition between process elements on different levels of detail.

In a real-world project it is also necessary to be able to trace logical connections between model elements as the relation between business activity and business rules or the relation between business requirement and business process. Document such relations to be able to understand the process of model creation later on.

Do: Document the relations between model elements to ensure traceability.

2.10 Use Tools which are easy to use

I already mentioned that you need tools to support your approach. There are many tools available in the market place. You will maybe not find a tool which covers all aspects of Enterprise Modeling. Try to use only a few tools, try to find tools which can be integrated to avoid media breaks.

A very important requirement for a visual modeling tool is "Ease of Use". I saw too many modeling tools where the main functionality, creating visual models, is really hard to use. Even if such tools have great other features as maybe simulation, the projects are frequently in trouble.

We saw some requirements a tool should fulfill already. I just summarize this here:

- A good modeling tool needs to be repository based and has to support reuse of model elements.
- A good modeling tool should support structuring models.
- A good business process modeling tool should support decomposition and composition of business processes.
- A good modeling tool should support sufficient documentation of model elements.
- A good modeling tool should support traceability to track relations between model elements and other models or between model elements.
- A good modeling tool should be able to integrate with other tools.
- A good modeling tool should offer different outputs as intranet pages, Word or PDF reports, and complex documents.

Do: Select modeling tools which are easy to use.

Do: Select a small set of tools, try to avoid media breaks.

Do: Select a modeling tool, which is repository based.

Do: Select a modeling tool, which support documentation and traceability.

Literature

Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0, OMG, August 2009, OMG Document Number: dtc/2009-08-14

Business Process Modeling Notation (BPMN), Version 1.2, OMG, January 2009, OMG Document Number: formal/2009-01-03

Ross, Ronald G., Business Rule Concepts, Third Edition, Business Rule Solutions, LLC, 2009