From BPMN Directly To Implementation - The Graphical Way

Heinz Lienhard and Bruno Bütler,
ivyTeam-SORECOGroup, Switzerland

BPMN process description must become what the name (Business Process Modeling Notation) actually implies – a true model that can be simulated, validated, and immediately turned into a real-time application.

To Model – that's it?

With BPMN a lot has been accomplished towards a unified description of Business Processes readily understandable to all business users, and also understandable (as stated in the introduction[1]) "to the technical developers responsible for implementing the technology that will perform those processes." But here the "understanding" is very likely to end. Experience over the past 10 years has shown one thing: Nice pictures of sophisticated business processes, per se, bring little value to the process. Although they may at first stimulate heated discussions, they are quickly forgotten and left to rot in some file cabinet. And giving those nice pictures to the IT department for implementation will more often than not lead to situations where business people (or "end users") who believe they understood the process will then find its implementation unintelligible.

Hence, the truly important step is the *implementing of* the process just as *transparently* and *comprehensively* as the *modeling* of the process in the first place. This means using graphical objects as much as possible, right up to implementation. It will be shown how this can be done. First, starting from the upper level BPMN diagrams, the model is being refined, and, by using an appropriate designer, the graphical symbols become true process building blocks (elements). With an integrated engine, the models built this way can be simulated and animated on any desired level. The final solution is obtained by the click of a button. Thus, process model and implemented workflow can be realized in one step:

"The BPMN model becomes the solution."

To go directly from model to solution has proven crucial to the success of real-life projects: When people can see how the solution will work – including on the level of implementation – they immediately become very involved and creative, and thereby participate constructively from the start.

Through example it can be shown how further development of BPMN might help to make process implementation much easier and, at the same time, more standardized.

[1] Working Draft BPMI.org

The case for BPMN

According to the working draft (ref. 1), “the primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.” The intent is to create a standard visual language that all process modelers will recognize and understand. This also means conformance requirements are necessary for the appearance of the BPMN graphical elements. At the same time there is room for extensions, especially for new markers (e.g., to define a new event), or for additional artifacts (without, of course, interfering with any current object shape). So far so good – with one reservation: after a very promising start the whole standard seems to get out of hand. The proposed standard now has almost 300 pages, which is hardly compatible with the primary goal stated above. Nevertheless, the “core element set” is very useful for capturing the gist of a business process.

It is further claimed that “BPMN creates a standardized bridge for the gap between business process design and process implementation” – to a certain point; but the gap (or divide) between model and final implementation remains. Business people like visualizing business processes, but they are certainly lost when confronted with the mapping into a so-called execution language, which is in essence a programming language. They can only hope that what they originally agreed upon (the nice looking BPMN diagram) will be realized by the implementation. And experience shows that the real problems usually start right here.

Obliterate The Divide

Smith and Fingar state in their book[2], “Don’t bridge the business-IT divide – Obliterate it.” That’s the solution – only they fail to really show how to do it. A lot of hand wringing about Pi-Calculus won’t do it. Besides, it is not even clear whether this mathematical underpinning is the best choice – See van der Aalst[3]. Certainly, a formal basis is necessary for the process elements, and tight integration of process modeling and workflow definition is a must to be able to go straight from model to executable solution[4].

Hence, why not present the implementation also in graphical form (BPMN) that can be simulated/animated, i.e., can be visualized to see how it works together with the necessary additional special elements (Tasks) for a running workflow; e.g., the interactions with the users, DB access, etc.? This way, transparency is obtained not only for the original process, but also for its realization. This is crucial, because, very often, hidden problems of a fine looking process model become apparent only during its execution. Now business people who do understand the process (and the implementation) can immediately help to find the correct solution.

[2] H. Smith and P. Fingar: “Business Process Management—The Third Wave”, 2003

[3] W.M.P. van der Aalst: “Pi calculus versus Petri nets…”, 2005

[4] *Workflow Handbook 2002:* “Workflow as a Web Application – the Grand Unification”

How to do it

The process elements as graphical objects will have to become more than just that. They do please the eye as graphical elements, but underneath they are well-defined mathematical objects enabling the process model to control the workflow's behavior. It has been shown[5] that by providing an appropriate palette of such implementation process elements, process models can be composed that allow immediate simulation of their behavior and can readily be turned into a real-time application on a properly prepared Web server, i.e., with an appropriate workflow server package installed. The BPMN model is preferably designed in a modeling and simulation tool such as has been described before (ref. 5) – or it is imported into such a tool. Once the graphical model has been entered one way or the other, we proceed with implementation through the following main steps:

- With the help of a Process Data Editor, structure and type of all data used by the process are specified.
- Simple BPMN objects, like the start object, are directly parameterized, e.g., with the link address that will actually start the process. Thus the BPMN object becomes an implementation process element.
- Non-atomic BPMN objects, like the independent (reusable) subprocess, will require the proper name (identifier) of the BPMN process defining the subprocess.
- Implementation details that have been hidden in a group are now expanded using the appropriate special elements (See examples in Fig. 4) and configured using the related assistants. All this can be done without any programming.
- Tasks with message flow can be implemented in various ways, again using special process elements; e.g., an eMail Task. It will automatically generate and send an email to an address either available from the process data or fetched from a data base. Other systems may be accessed via WebService Call Elements (See Ref. 5).
- Animate and test the obtained model within the design tool. The animation may be observed on any desired hierarchical level: A data object, represented by a red dot, moves through the model. Any forms for user interaction are displayed (preferably on one side of the screen) within the chosen Internet browser – this way interaction with the model and the actions of the model can be pursued simultaneously.
- Upload the model onto the Web server on which the server package has been installed.
- Run the implementation.

Evidently, the model design tool mentioned above has to offer a rich enough palette of special BPMN objects to directly implement the process (which is the case for Xpert.ivy from ivyTeam-SORECOGroup). With such an approach neither BPEL nor anything like it is needed: The obtained workflow is perfectly capable of orchestrating any Web services

[5] *Workflow Handbook 2003*, "Web Services and Workflow—a Unified Approach"

(or choreographing them, to use this even more misplaced term – Why can't IT people not stick to their turf?)

A practical example

With a simple, yet practical, example from eGovernment (from the authors' country), the procedure expounded before makes it clear how elegantly processes specified in BPMN can be implemented as workflows. The Swiss government decided to model future eGovernment processes in BPMN (or, rather in a useful subset of it) and to provide an exemplary process called "Entry with Working Permit." Through this process foreigners wishing to live and work in Switzerland are granted (or refused) an official permit. The process has been made public to help communities, cantons (states), and cities in modeling such government processes. Incidentally, it was this process that originally looked very different in various cantons until it was carefully remodeled in BPMN. Only then it became apparent that it was much the same process everywhere (See Fig. 1).

The example contains two pools – one is the pool of applicants (usually the future employer of a foreigner) and the administration receiving and processing the application. The second pool contains three lanes specifying who is responsible for what activity. Since the actual communities may be organized in different ways – the norm in a federalist country – more or less abstract roles have been chosen, which will be named differently depending on the concrete local administration. Without such a standard as BPMN, finding a common process would not have been possible.

The applicant is required to send certain documents with the application. When received by the administration, the Receiving Office will start the workflow (Fig. 2). Included in the expansion is an independent subprocess in which the documents are checked for completeness (Fig. 3). If incomplete, the flow is directed to the task, "Ask for missing documents," where a message is sent, followed by the intermediate event (double circle), which will be triggered once the missing documents have been received and the case is resumed via work-list of the person having the proper role. The overall process should be easy to understand.

Two comments: RIPOL is a system where people wanted by police are registered; ZAR is a central register for foreigners. This year it will become accessible via Web service Interface.

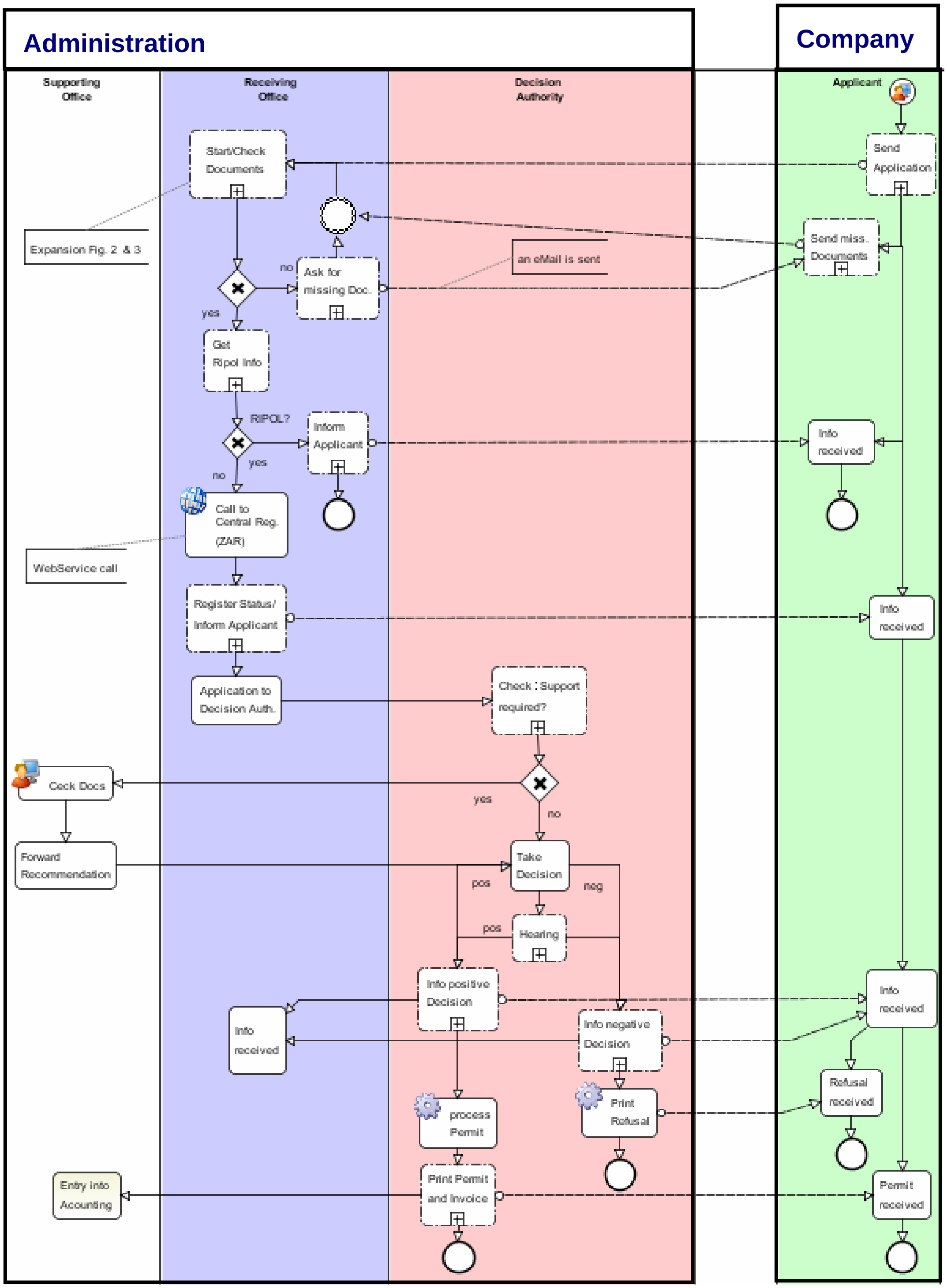
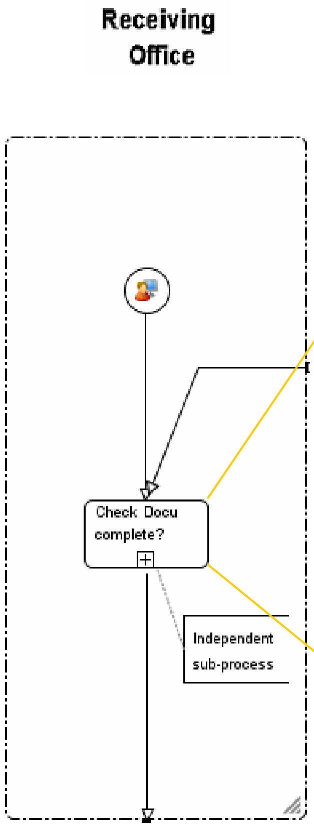


Figure 1. eGovernment process “Entry with Working Permit”



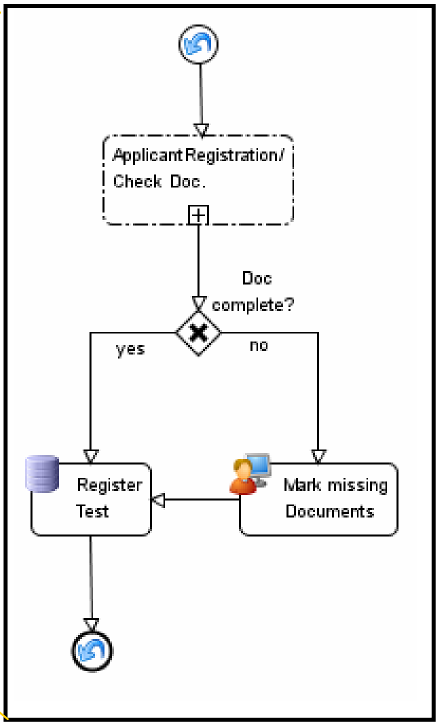


Figure 2. Expanded group “Start/Check Documents.”

Figure 3. Independent sub-process “Check Docu complete?”

The group called “Start/Check Documents” of Figure 1 is expanded in Figure 2. The expansion of the subprocess contained in this group is shown in Figure 3.

The task “Mark missing Documents” (Fig. 3) requires a special process element: A Dialog Task that will generate a form in the user’s browser to enter data. This element comes with an assistant that helps to create such forms efficiently and without any programming (Fig. 4).

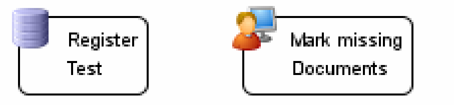
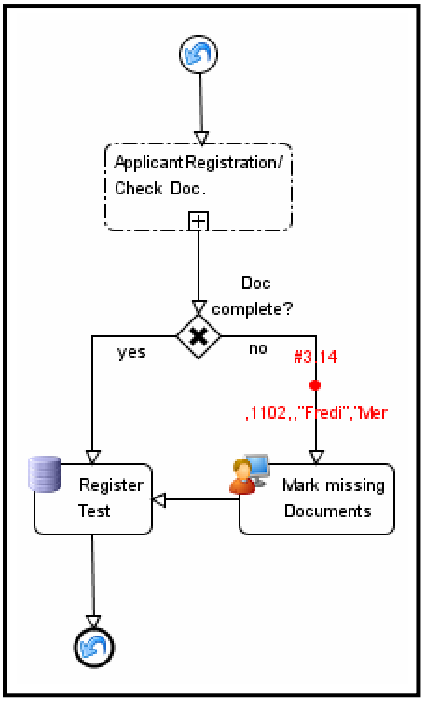
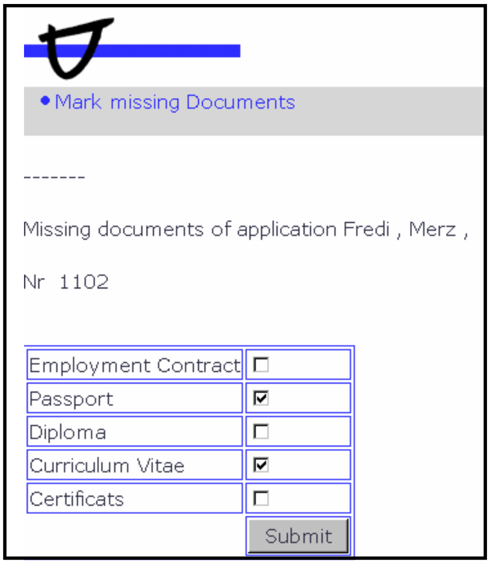


Figure 4. Examples of special elements: DB and Dialog Tasks



Now a sequence of snapshots should give an idea of what can be seen during simulation/animation. In Figure 5 the independent sub-process of Figure 3 is being simulated: The data object (red dot) is about to enter the special Dialog Task. Two data items are made visible – number and name of the applicant. An instant later this object will enter the Dialog Task thereby generating the form exhibited in Figure 6. The process now waits until the form's "Submit" button is pressed by the user after having marked the corresponding documents. This information will be forwarded in the process for future use (e.g., by the eMail Element – See below).

Figure 5. Independent subprocess during simulation.



• Mark missing Documents

Missing documents of application Fredi , Merz ,

Nr 1102

Employment Contract	☐
Passport	☑
Diploma	☐
Curriculum Vitae	☑
Certificats	☐
	Submit

Figure 6. The form is displayed (with the logo from Canton Zug) in the user's browser.

Going back to the top level, we see how the data object is about to enter the task that will send a message to the applicant (partial view of top level in Fig. 7). The object carries the result on the missing documents, which now will be used in the email. The unfolded group is given on the right side of Figure 7. Two special process elements are needed from the tool's palette – the Data Base Task to get the email address of the applicant and the eMail Task to send an automatic mail to him.

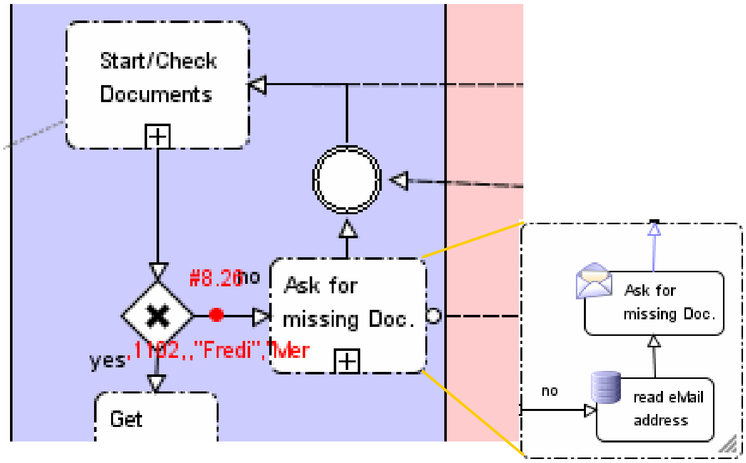


Figure 7. Clipping of top level simulation with expanded group.

Once all required documents have been received from the applicant, the process moves on, down the lane "Receiving Office." The refinement of the task, "Call to Central Reg.," can again be done using a single process element – the WebService Element (Ref. 5), which can be readily configured to access the central register, etc.

The implementation of this eGovernment process example took roughly one day and was presented to the administration. Depending on further needs and requirements, another few days might be necessary to obtain a final solution.

What about XPDL

When the (implemented) model has been validated, it can be uploaded onto the server, as described above. For this the model is currently encoded into XML format. This format will be changed in the next version of the tool, adapting it to XPDL. Hence, the approach presented here will also become an effective way to map BPMN to the XPDL process exchange format.

Benefits and outlook

One group of people explicitly targeted by the BPMN standard – “the business people who will manage and monitor the processes” – has not yet been addressed here. But these people will likely profit the most from the possibility of rapid adaptation of processes *and* their transparent implementation. It has been explained elsewhere[6,7] how process management becomes much more efficient if processes are properly designed in the first place, e.g., by making judicious use of business rules in combination with catalogs of independent subprocesses, and implementing them in the way proposed in this chapter.

How to proceed with BPMN? Actually, the standard tends towards the verbose. A modest proposal follows:

For modeling, go back to the graphical core object set of BPMN, and then provide the possibility of an expandable set of standard implementation elements (the special task above) with well-defined semantics. Although the design tool mentioned above does this already, such an extension of the standard would bring transparency in the process implementations to the various users. It would help to achieve what has been claimed in the introduction:

"The BPMN model becomes the solution."

Acknowledgement: The paper originally appeared in the Workflow Handbook 2006 published by Future Strategies Inc., in collaboration with the Workflow Management Coalition, and is used with permission. Some revisions have been made to take into account of the latest developments of the BPMN Editor/Workflow Designer mentioned in the text.

[6] Pallab Saha (Ed.) “Enterprise Systems Architecture Practice” (Chapter “EA as a Meta- Process”) to be published by IDEA GROUP, INC, Hershey, USA
[7] *Workflow Handbook 2005*, “Workflow and Business Rules—a Common Approach”