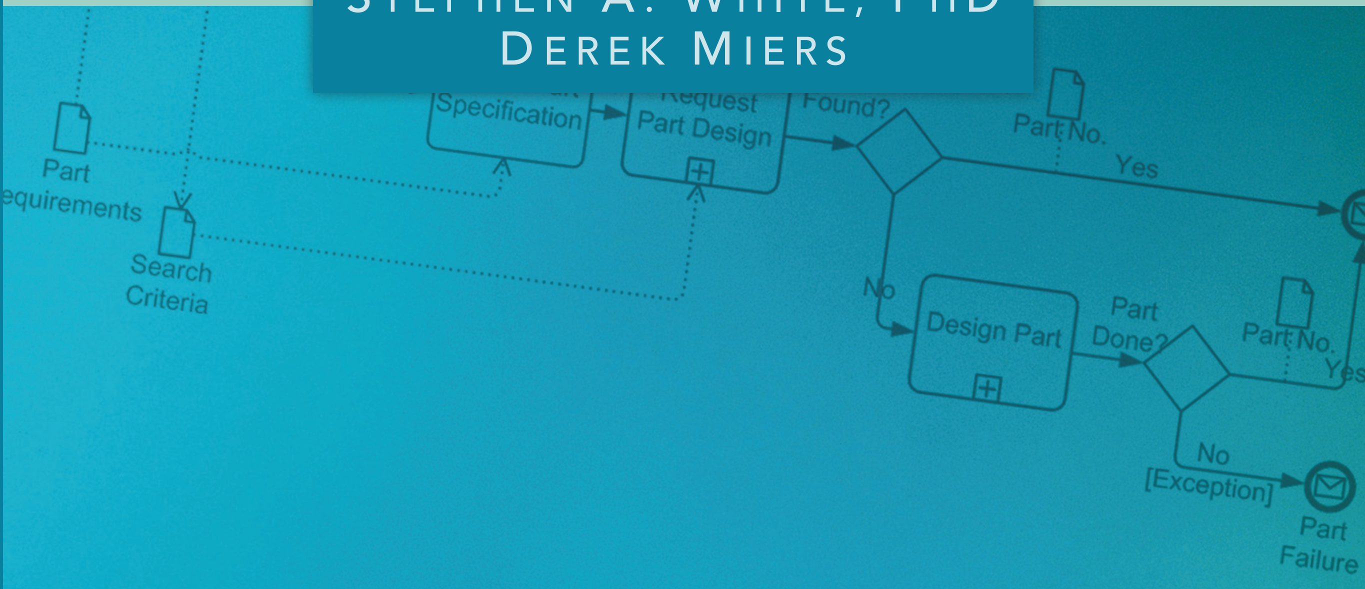
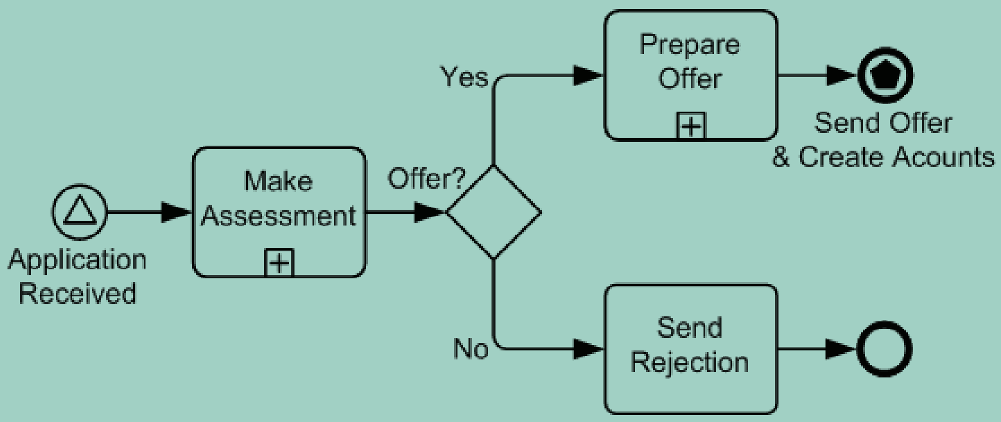
Stephen A. White, PhD
Derek Miers



BPMN

MODELING AND REFERENCE GUIDE



Develop rigorous yet understandable graphical representations of business processes

BPMN Modeling and Reference Guide

Stephen A. White, PhD   Derek Miers

BPMN
Modeling
and Reference Guide

Retail Price: $39.95

Buy securely online here:

PRINT EDITION
http://store.futstrat.com/servlet/Detail?no=49

DIGITAL EDITION
(download book immediately 4MB PDF, including Exercises and Answers)
http://store.futstrat.com/servlet/Detail?no=63

BPMN Edición digital en español
BPMN Guía de Referencia y Modelado US$49.95
The only BPMN book in Spanish.
http://store.futstrat.com/servlet/Detail?no=68

If you prefer to use your Amazon.com account, use this link:
http://tinyurl.com/yfs64jl
(PRINT EDITION only. For the Digital Edition or Spanish Edition, please order from our website.)

BPMN Modeling and Reference Guide

UNDERSTANDING AND USING BPMN

Develop rigorous yet understandable graphical
representations of business processes

STEPHEN A. WHITE, PHD
DEREK MIERS

Future Strategies Inc.
Lighthouse Point, Florida, USA

BPMN Modeling and Reference Guide:
Understanding and Using BPMN

Copyright © 2008 by Future Strategies Inc.

ISBN10: 0-9777527-2-0
ISBN13: 978-0-9777527-2-0

Published by Future Strategies Inc., Book Division
2436 North Federal Highway, #374,
Lighthouse Point, FL 33064 USA
954.782.3376 / 954.782.6365 fax
www.FutStrat.com — books@FutStrat.com
Cover: Hara Allison www.smallagencybigideas.com

All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse should not be regarded as intent to infringe on the property of others. The Publisher recognizes and respects all marks used by companies, manufacturers and developers as a means to distinguish their products. Neither the authors, the editors nor Future Strategies Inc., accept any responsibility or liability for loss or damage occasioned to any person or property through using the material, instructions, methods, or ideas contained herein, or acting or refraining from acting as a result of such use. The authors and the publisher expressly disclaim all implied warrantees, including merchantability or fitness for any particular purpose. There will be no duty on the authors or Publisher to correct any errors or defects in the software.

Publisher's Cataloging-in-Publication Data
ISBN: 978-0-9777527-2-0
Library of Congress Control Number: 2008932799

BPMN Modeling and Reference Guide: Understanding and Using BPMN
/Stephen A. White, PhD., Derek Miers
p. cm.

Includes bibliographical references and appendices.
1. Business Process Management. 2. Process Modeling. 3. Enterprise Architecture. 4. Notation Standard. 5. Workflow. 6. Process Analysis

White, Stephen A.; Miers, Derek

Table of Contents

Chapter 4. Modeling Issues

Abstract: *The purpose of this chapter is to discuss some of the issues associated with Process modeling in general, and to identify some of the challenges in dealing with these issues. The associated Appendix (Techniques for Process Architecture on page 199) discusses several of the available approaches that can help the modeler identify an appropriate Process Architecture.*[5]

"All Models are Wrong, Some are Useful"

This quotation, variously attributed to Edwards Deming but actually originating from the lesser-known Charles Box,[6] describes the predicament in which modelers find themselves. There are usually a great many ways of modeling a desired behavior, at any number of levels of precision.

Key Point: *Many people assume that there is always a correct model (and that somehow other models are wrong) however, there is seldom only one correct model. On the other hand, models might be <u>invalid</u> (in that they incorrectly use a given notation).*

Moreover, there is often far more potential detail to capture than is necessary. If we were to model how one goes about making a cup of tea, then a single Activity might be sufficient. Alternatively, one could describe the need to first boil the water, place a tea bag in a cup and optionally add milk. But what if we liked to brew the tea for several people using a teapot and tea leaves, or should we include the steps involved in filling the kettle, or adding sugar. The modeler is always making decisions about what to include and what leave out. So one needs to maintain a perspective about the uses of the model and who will interpret it.

If the audiences (those who will read and interpret the model) are not interested in the fine detail, then do not include it in the model. In other situations, such as where the model will

[5] While this is outside the scope of the BPMN standard itself, this is of interest to modelers as they tend to assume that somehow BPMN will help them decide on what processes exist for a given domain.

[6] Charles Box first used it as a heading in a book chapter in 1979—Citation: Box, G.E.P., *Robustness in the Strategy of Scientific Model Building*, in Robustness in Statistics, R.L. Launer and G.N. Wilkinson, Editors. 1979, Academic Press: New York.

support execution on a BPM Suite or where simulation is the objective, then significant detail is normally required.

At the beginning of one of our workshops, we go through a very simple exercise. We ask delegates to brainstorm all the things that they would want represent on process models. It is not long before we have filled a couple of white boards—activities, flow, inputs, outputs, responsibilities, costs, locations, quality, rules, interactions, escalation, etc. Asked if they would want all of these dimensions to appear on a single process, delegates suddenly realize it is a question of removing things from the models to make them useful.

Key Point: *The modeler is constantly making modeling decisions about the purpose of the model and the intended audience.*[7]

An anecdotal story drives home the point. During the days of Business Process Reengineering (BPR was sometimes referred to as Bigger People Reductions), a major chemicals giant employed one of the leading consulting firms to assist with the reengineering of their North American sales process. After several months of work, a presentation was staged for the main board (as this was a highly important project). On one side of the meeting room was an eighty foot flow diagram (the As Is model). On the other wall, a sixty foot flow diagram of the To Be process. The then-Chairman allowed the Consultancy Partner to complete his presentation before asking a very simple question. "Is that a good process and if so, please explain why." And therein lay the core of the problem. The detail delivered was wholly inappropriate for the intended audience.

Here are some traits of a good model:[8]

- **Salient**—Since no model can represent everything, it must selectively represent those things that are most relevant to the task at hand.
- **Accurate**—The model should precisely encode the actual state of affairs and not an erroneous or biased view.
- **Complete yet Parsimonious**—The model should be as simple as possible, but no simpler.[9]

[7] Some vocal modelers seem to feel that notation should provide only one way of representing any particular problem. But this attitude flies in the face of reality and expecting only one possible model for a given scenario is unrealistic; all models are a compromise. BPMN often provides a range of functionality to facilitate different modelling purposes and styles.

[8] Marshal Clemens of consultancy firm, Idiagram, offers some excellent guidance on the features that models should exhibit. He is not discussing BPMN, but many of the points are still relevant. http://www.idiagram.com/ideas/models.html

- **Understandable**—Once we perceive the model we must be able to make sense of it; it shouldn't be too complicated or unfamiliar for us to understand.

Clemens goes on to point to some of the evolution and adaptability issues around modeling. "As all models are, to some degree, inaccurate, irrelevant, mistaken, time-sensitive etc., they should be open to recursive revision to reflect new data, our growing understanding, or our evolving needs."

In the end, models need to be useful. Clemens continues, "Usefulness is the sum of the above properties and the degree to which they combine to promote understanding and effective action. It is important to note that the most accurate, or the most complete, or the most elegant model is not necessarily the most useful. All models are incomplete. All models are a compromise. The model maker's art lies in making those shrewd trade-offs that will render the model most useful to the problem at hand."

Key Point: *In order to be useful, all models selectively represent some elements of the real world. The modeler excludes different dimensions of the domain (in order to achieve the modeling goals).*

How Many Processes, Where Do They Fit?

The temptation is always to leap straight in and start modeling. Yet a more considered approach normally pays significant dividends.

The real problem is that as people begin to describe how things happen in an area of their organization, they assume that it is all one big process. We often see it in our workshops. Students try to connect everything up together into one amorphous process description that captures every possible permutation.

Key Point: *Very often, it is inordinately difficult to model one "end-to-end" process for a given business problem. And even if it were possible, it is challenging to make that model flexible and adaptable.*

It is usually far better to break up a given domain problem into a number of discrete "chunks," that working together solve the problem. So the issue becomes one of how to come up with the

[9] Here he is paraphrasing Einstein.

right chunks. But when looking for techniques, one finds remarkably few.

For a wider discussion on the various approaches for organizing, scoping models, see the Appendix “Techniques for Process Architecture” on page 199. Here we outline a set of approaches that, between them, provide a translation from the business strategy level right through to a robust process architecture (independent of the reporting structure of the organization). Potentially, these techniques could extend into a stack of IT services (as part of a Service Oriented Architecture).

The point is that BPMN is “methodology agnostic.” Organizations typically have a preferred methodology for capturing and developing their business processes. It is not the role of BPMN to dictate how business process information is collected or how modeling projects are undertaken. Therefore, BPMN supports multiple methodologies (being as simple or complex as it needs to be). It does not specify the level of detail for models—the modeler, modeling tool, or organization makes these decisions. Indeed, as we will see with process modeling in general; usually there exists many different ways of modeling the same situation, with any number of different levels of detail.

Key Point: *BPMN does not provide any advice on how to structure a domain or come up with an appropriate architecture for a given area. Yet it provides capabilities that can support many different methods.*

Dealing with Complexity in BPMN

So as we can see from the above, business processes can become complex—very complex (covered in more depth in the Appendix). However, most developers and readers of process models want a simple, graphical language for depicting Business Processes. In fact, the majority of all process models are simple flow charts (activity boxes, decision diamonds, and the connectors between them). At the same time, modelers need enough flexibility to represent further levels of complexity if they needed.

The objective of most process modeling projects is to document (understand) and analyze an organizations key business processes. Yet these same models can then become the basis for a more detailed set of Process descriptions for other uses. Elaborated and built upon with further detail, they might then become executable (in a BPM Suite or workflow tool).

For example, a rather simplistic model (originally developed for a business model discussion), may end up being adapted for use in establishing appropriate partner relationships (defining the interfaces), which is then further embellished and adapted by both parties to support their respective process execution environments.

Since each company or modeler may want to show different levels or areas of complexity, the notation needs enough flexibility to handle virtually all possible business situations or modeling requirements. But the problem is that such a modeling notation, one that is capable of depicting all business situations, is no longer simple, it is complex.

This issue points to the dynamic tension that exists between the two primary goals of BPMN:

- On the one hand, ease of use for business user and business analyst;
- And on the other, executable processes.

To meet the requirements of the first goal, BPMN is structured with a small set of elements (e.g., Activities, Events, and Gateways) that have distinctive shapes (e.g., rectangle, circle, and diamond). The small set of main elements supports the simplicity and readability of the models.

To meet the requirements of the second goal, the main elements are specialized for specific purposes, each of which carry further information and/or supported with more elements to allow the modeling of the required behavior. In addition, the underlying semantic structure of BPMN must be rigorous, containing information that enables the generation of valid BPEL; or at least set the stage for other tools to complete the development and deployment.[10]

The BPMN specification includes a lot of information and capabilities that make it look complicated. However, it is unlikely that the business analyst or end user needs most of these capabilities (as they relate to the execution semantics). In this book, we will point out the core BPMN elements that should appeal to the Business Analyst, while also providing thorough descriptions of the more advanced BPMN elements.

[10] With a rigorous definition of the semantics of a BPMN model, some BPM Suites are capable of executing the Process model directly, without further translation to an intermediate language such as BPEL. With the emergence of BPMN 2.0 this capability will be further enhanced as the underlying semantics become yet more rigorous.

Key Point: *While the BPMN modeling technique may appear a little daunting to the uninitiated, it is only as complex as it needs to be in order to support both ease of use for the business analyst and end user; and at the same time, enable elaboration of models to support process execution.*

Chapter 5. Scenario-Based BPMN Introduction

Abstract: *This chapter provides the reader with a gradual introduction to the BPMN specification, taking an easily understood scenario and then slowly building upon it, bringing in BPMN functionality within that described context.*

Designed for those coming to BPMN for the first time, it allows them to familiarize themselves with the core features of the Notation without being overwhelmed by the complexity of some of the more esoteric aspects.

Most of the functionality is limited to the "core" set of BPMN elements with which a Business Analyst should be familiar. This concept of the core set is expanded upon in the reference section.

Building out a Process with BPMN

The central scenario used within this chapter revolves around a fictitious organization Mortgage Co. They take applications from potential customers, make an assessment whether or not to offer the mortgage, and then either reject the application or make the offer (see Figure 5-1).[11]

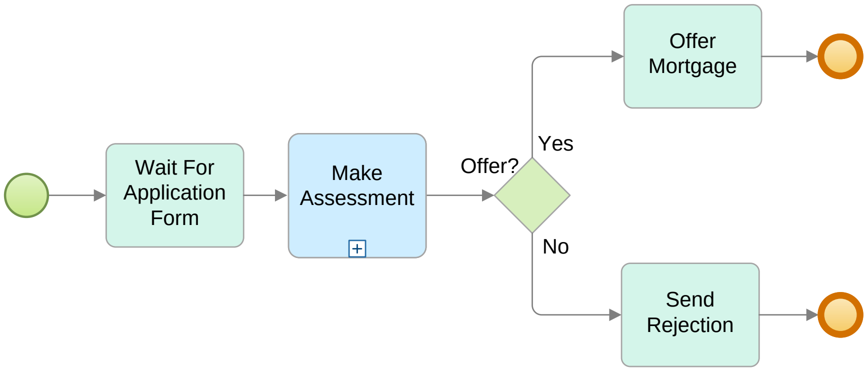


Figure 5-1—The underlying mortgage offer scenario

Clearly, this is a rather simplistic picture of how such a process might operate. But it will suffice in providing the backdrop for us to introduce the functionality of BPMN. Through

[11] All paragraphs that build on the underlying scenario will share this font style (indented slightly and italics).

the remainder of this part of the book, we will systematically build on that underlying scenario, embellishing the story and bringing in the appropriate BPMN modeling features to represent the desired behavior.[12]

The Process begins on the left with a Start Event (thin line circle), with two Activities (rounded rectangles) connected to the Start Event with Sequence Flow (the arrows). The first Activity is a Task and the second represents a Sub-Process. Following a *decision*, represented by the diamond (called an Exclusive Gateway), the Process then branches to either "Offer Mortgage" or "Send Rejection" (both represented here as simple Tasks). Both branches lead to an End Event (thick circle).

Start Events represent the places that a Process can *start*, End Events represent different *results*, some of which might be desired and others not. An Exclusive Gateway represents a binary decision—only one *outgoing* Sequence Flow can evaluate to *true*. For the purposes of this model, the three Tasks represent simple "atomic" steps, whereas the *collapsed* Sub-Process has a further level of detail.

More details on the elements introduced are available in the BPMN Reference Section:

Start Events on page 87

- Tasks on page 68.
- Sub-Processes on page 69.
- Exclusive Gateways on page 139.
- End Events on page 127.
- Sequence Flow on page 173.

Setting Timers

> *Now, let us assume that we want to represent the fact that our potential customer contacted Mortgage Co to ask for a mortgage application form. For the moment, we will not worry about precisely how they contacted the company, but let us assume it was a "message" of some sort. Further, we want to set a clock running to send them a reminder after seven days if Mortgage Co does not receive their application form back (see Figure 5-2).*

[12] We refer to the graphical elements of BPMN with Initial Capitals. Where an important BPMN concept is referenced (that is not a graphical element), we have used *italics* within the sentence.

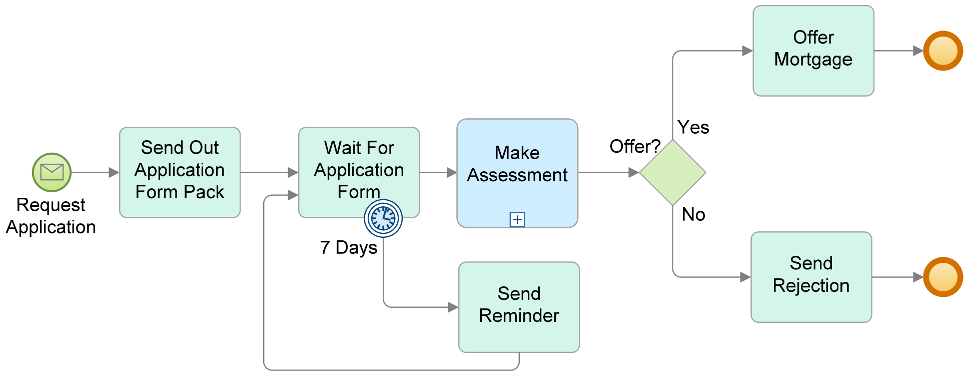


Figure 5-2—A Message Start Event and a Timer Intermediate Event are introduced

The Process now begins with a Message Start Event representing the message received by Mortgage Co who then sends out the application form; a timer is placed on the waiting task to interrupt it and send a reminder before *looping* back again to wait for the application form again.

There are many types of Start Events in BPMN; here we have used a Message Start Event to indicate how this Process begins. Intermediate Events placed on the boundary of a Task means that if the Event fires, then it will interrupt the Task and send the Process down its *outgoing* Sequence Flow. If the Task completes before the Intermediate Event fires, then the Process moves on normally (following the *normal flow* of the Process). The loop is created explicitly with Sequence Flow although, as we will discover later, there are alternatives (i.e., use a Loop Task).

More details on the elements introduced are available in the BPMN Reference Section:

- Message Start Events on page 91.
- Interrupting Activities with Events on page 101.
- Timer Intermediate Events on page 105.
- Looping on page 77.

There is another way to model this scenario using a Sub-Process for the send out application form and wait for the response Figure 5-3.

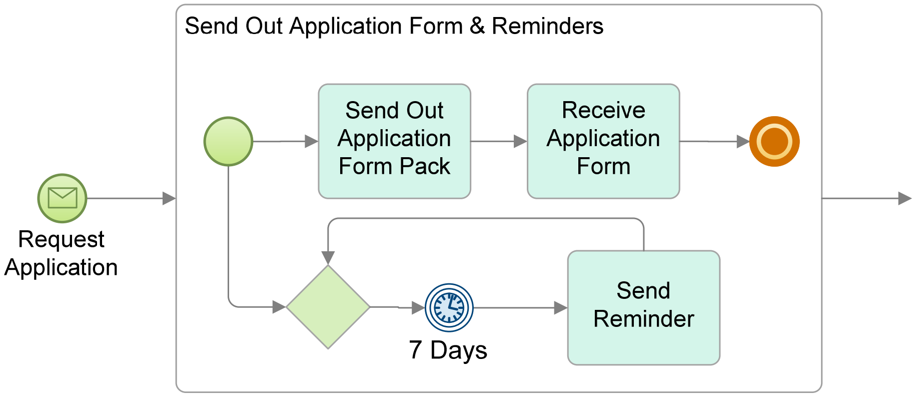

Send Out Application Form & Reminders
Send Out Application Form Pack
Receive Application Form
Request Application
Send Reminder
7 Days

Figure 5-3—Using a Sub-Process to represent the application form and reminders

The Timer Intermediate Event shown in line with the Sequence Flow triggers immediately the Sub-Process begins (the Sub-Process is shown in its *expanded* form). It waits for seven days before that thread of activity moves to the “Send Reminder” Task before looping back to wait for another seven days. When an Intermediate Event is used in line (as in this case), then it can have only one *incoming* and one *outgoing* Sequence Flow. Therefore, merging the *incoming* Sequence Flow before the Timer Intermediate Event requires an Exclusive Gateway. When *merging* Sequence Flow, an Exclusive Gateway immediately passes through any *incoming* Sequence Flow so in this case it serves to clean up the Sequence Flow (but does not represent any sort of delay).

Of course, other *flow objects* (Activities or Gateways) can normally have multiple *incoming* and *outgoing* Sequence Flow. While the Sub-Process could have included a Parallel Gateway to create the split (see Figure 5-4), it is unnecessary as the Sequence Flow does not require control. Figure 5-3 and Figure 5-4 describe exactly the same behavior. A general rule is that Gateways are only required where Sequence Flow requires *control*.

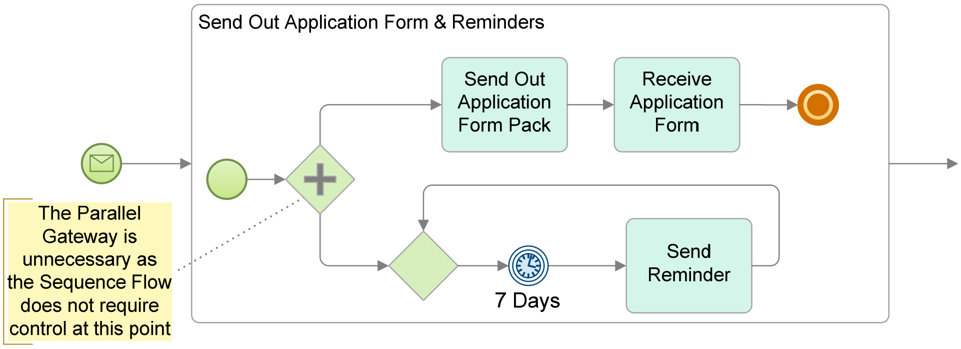

Send Out Application Form & Reminders
Send Out Application Form Pack
Receive Application Form
Send Reminder
7 Days
The Parallel Gateway is unnecessary as the Sequence Flow does not require control at this point

Figure 5-4—Using a Parallel Gateway is unnecessary

The Sub-Process finishes with a Terminate End Event. The Terminate End Event causes the immediate cessation of the Process on its current level (and below) even if there is still ongoing activity. Effectively, it kills off the reminder *loop*.

More details on the elements introduced are available in the BPMN Reference Section:

- Timer Intermediate Events on page 105.
- Terminate End Event on page 131.
- Parallel Gateways on page 147.
- Text Annotations on page 171.

Exercise One

Try modeling this process; it will help ensure that the techniques discussed so far sink in:

> Every weekday morning, the database is backed up and then it is checked to see whether the "Account Defaulter" table has new records. If no new records are found, then the process should check the CRM system to see whether new returns have been filed. If new returns exist, then register all defaulting accounts and customers. If the defaulting client codes have not been previously advised, produce another table of defaulting accounts and send to account management. All of this must be completed by 2:30 pm, if it is not, then an alert should be sent to the supervisor. Once the new defaulting account report has been completed, check the CRM system to see whether new returns have been filed. If new returns have been filed, reconcile with the existing account defaulters table. This must be completed by 4:00 pm otherwise a supervisor should be sent a message.

Looping

So far, the *loop* is expressed using explicit Sequence Flow coming back to an earlier part of the Process. BPMN provides another mechanism to represent this sort of behavior—the Loop Task (see Figure 5-5). A Loop Task has a small semi-circular arrow that curls back upon itself.

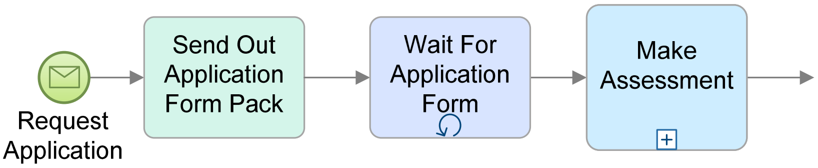


Figure 5-5--A simple Loop Task

It is possible to set BPMN attributes to support sophisticated looping behavior.[13] This is required to support the necessary complexity required by simulation and process execution environments. These aspects are discussed fully in the BPMN Reference Section.

> *Now clearly, it does not make much sense to endlessly loop back to wait for an application form that may never arrive. So after two such reminders, Mortgage Co has decided to cancel the application and archive the details.*

There is another way of setting the loop counter in Figure 5-6. Instead of using a graphically modeled "Set Loop Counter" Task, the "Send Reminder" Task could set an *assignment* at the level of the attributes. Although invisible, an annotation could then highlight its existence.

It is worth noting that the explicit Sequence Flow *loop* cannot cycle back to the Start Event. Indeed, Start Events cannot have *incoming* Sequence Flow. The *loop* can only go back as far the first Task.

[13] Looping and other element attributes store information about the Process that is not shown graphically.

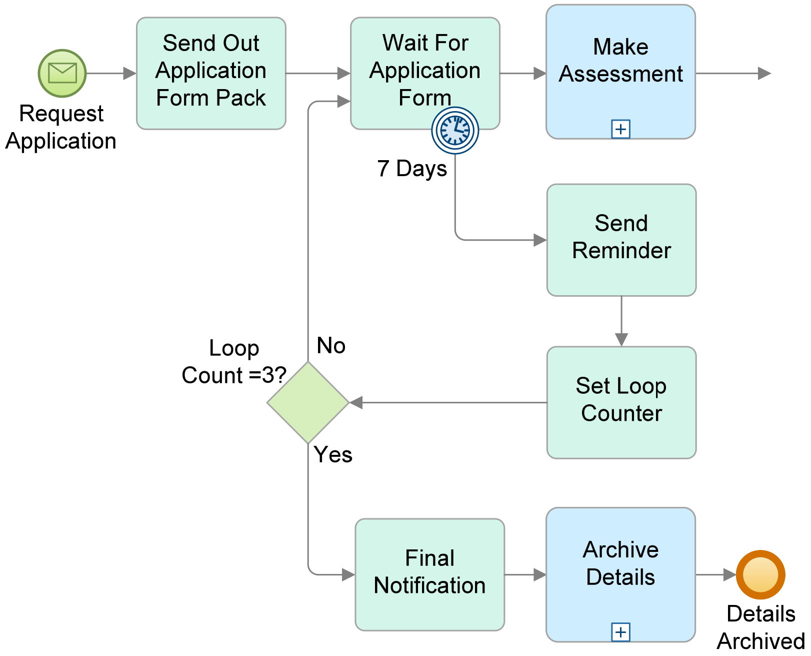


Figure 5-6—A loop counter is set and after two iterations, the details are archived and the Process ends

More details on the elements introduced are available in the BPMN Reference Section:

- Looping on page 77.

Decisions Based On Events

> *Of course, if the customer never sends back their application form, then the process will never get to the assessment phase. But what if the customer does let Mortgage Co know that they do not wish to proceed with the mortgage? The model in Figure 5-6 does not adequately represent this subtly different scenario.*
>
> *Now, after sending the application pack, Mortgage Co waits for one of three different things to happen. Either they receive the application (it moves on to the "Make Assessment" Task), or they are notified that the customer does not wish to proceed (in which case "Archive Details"), or after 7 days a reminder is sent (twice before sending a final advice and archiving the details).*

While it is possible to model such a scenario using Activities, Sequence Flow and Exclusive Gateways, the model would become very messy and convoluted. There is another way of modeling this situation, making use of an Event-Based Exclusive Gateway (see Figure 5-7).

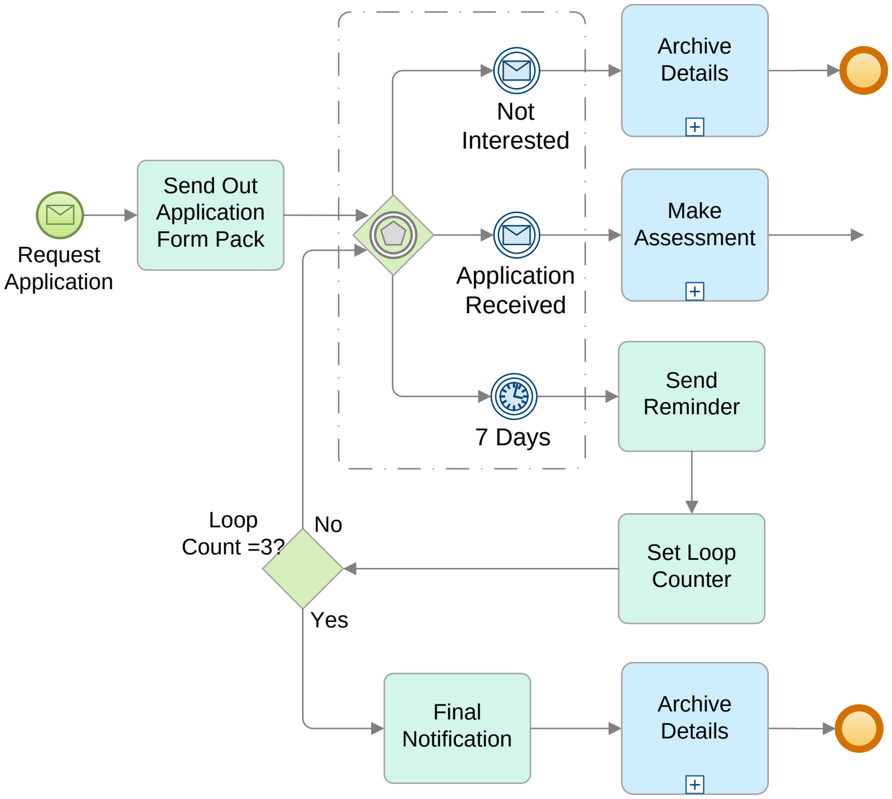


Figure 5-7—Using an Event-Based Exclusive Gateway

The Event-Based Exclusive Gateway (or informally, the Event Gateway) and its following Intermediate Events are regarded as a whole (the dot-dashed line around them is a BPMN Group used for emphasis only). To differentiate it from other Gateways, the Event Gateway reuses the Multiple Intermediate Event marker in the center of the diamond. Effectively, the Gateway waits for one the subsequent Events to occur. Either a *message* is received (Message Intermediate Event) indicating the customer is "Not Interested" or the "Application Received" Message Intermediate Event occurs (and the Process can progress), or the *timer* goes off and the reminder *loop* is initiated. Another Sub-Process could represent the reminder *loop*.

Notice that the "Archive Details" *collapsed* Sub-Process appears twice on the diagram. This Sub-Process is designed as a *reusable* Sub-Process. It might appear in other Processes outside the scope of this. Effectively, it represents a stand-alone Process referenced by this one. Of course, one could reorganize the diagram to use only one Activity on this model.

More details on each element introduced are available in the BPMN Reference Section:

- Event-Based Exclusive Gateways on page 144.
- Message Intermediate Events on page 110.
- Groups on page 168.
- Multiple Intermediate Events on page 126.

Meeting SLAs

> *Now let us assume that Mortgage Co receives the application form back and they have decided to institute a Service Level Agreement with their customers. They are now promising to respond with an offer or rejection within 14 days from the date of receipt of an application form. In support of this, the Process should alert the manager after 10 days if it has not completed, and then every day thereafter. Also, they need to archive the details if the decision was to reject the application (before the end of the Process).*

Thinking about the alert, the first temptation is probably to use a Sub-Process and then attach a Timer Intermediate Event to its border to create the alert (similar to Figure 5-2 on page 41). The problem with this approach is that it will *interrupt* the work of the Sub-Process, and a *loop* back to the beginning would cause the work to start again (not the desired behavior). The work should not stop just to raise an alert to the er. Figure 5-8 shows one approach to solving this problem.

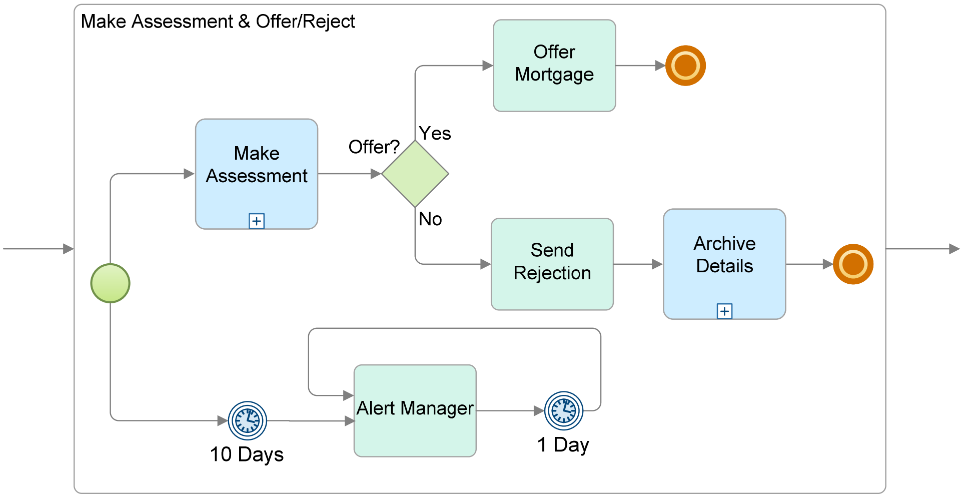


Figure 5-8—One approach to the non-interrupt alert problem

A separate Process *path* (or *thread*) with a Timer Intermediate Event linked to the Start Event of the Sub-Process is one approach to create a non-interrupting alert. The *timer* kicks in after 10 days if the work of the other thread has not finished—if that work is completed, then one or other of the Terminate End Events will kill off the *timer*. Effectively, a *race condition* occurs between these two strands of the process. Once the "Alert Manager" Task has occurred, it waits another day before looping back.

Representing Roles in Processes

The "Alert Manger" Task in Figure 5-8 above seems to imply that the manager receives a *message*. However, *messages* have a special importance in BPMN. Message Flow can only move between separate *participants* in a business-to-business situation. Each *participant* operates a separate Process represented by Pools. Message Flow coordinates the Processes of each *participant.*

Essentially, a Process exists within a single Pool. Labeled boxes display the Pool; they also have square corners as opposed to Tasks and Sub-Processes, which have rounded corners. BPMN uses Pools when representing the interaction between an organization and *participants* outside of its control. Within a company, a single Pool covers its own internal operations—it is only when it interacts with external *participants* that additional Pools are required.[14]

> *For example, in our Mortgage Co, the Credit Agency (and the Customer) would have a separate Pool (assuming one was trying to represent the interactions between the parties).*

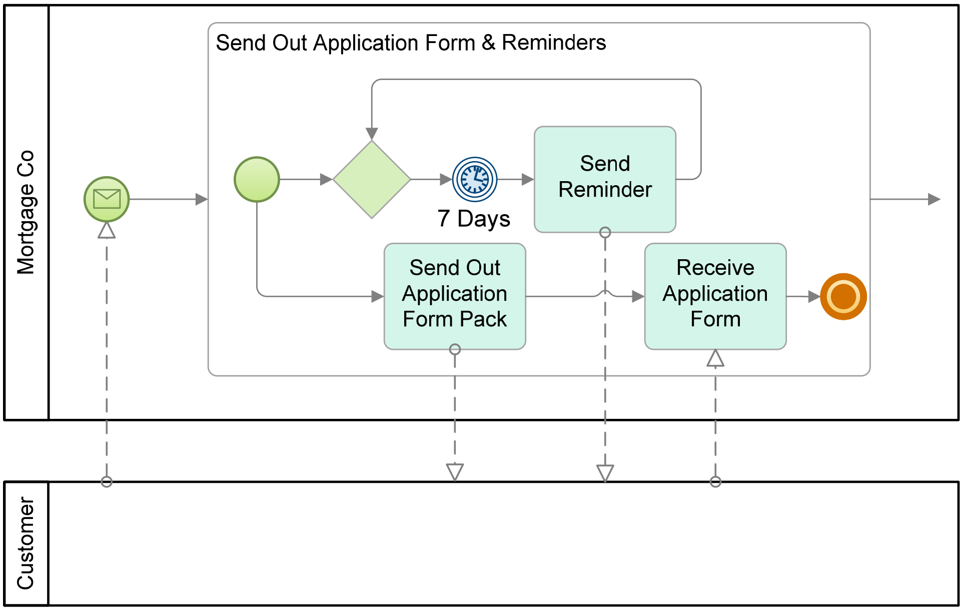

Mortgage Co
Send Out Application Form & Reminders
7 Days
Send Reminder
Send Out Application Form Pack
Receive Application Form
Customer

Figure 5-9—Representing the customer in a separate Pool

Message Flow cannot communicate between Tasks inside a single Pool—that is what Sequence Flow and *data flow* (as we shall see below) does. It moves the Process from one Activity to another. In this example (see Figure 5-9) the "Customer" Pool is shown interacting with a fragment of the "Mortgage Co" Process.

Mortgage Co does do not know the Customer's internal Process. Hence, the representation for the Customer is a "Black Box Pool." Within the Mortgage Co's Pool, the Message Start Event receives an *incoming message* from the Customer, which triggers the Sub-Process. A race condition starts between the two threads of the Sub-Process.

[14] Separate Pools might be used where an organization had several independent business units that were collaborating. In such a situation, each business unit would not necessarily know the internal operations of the others, yet would need to indentify the interfaces between them.

Additional BPM Resources

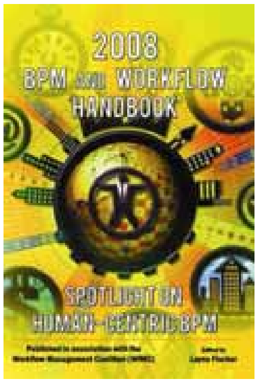
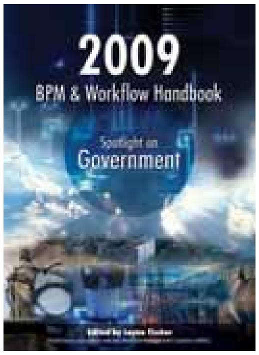
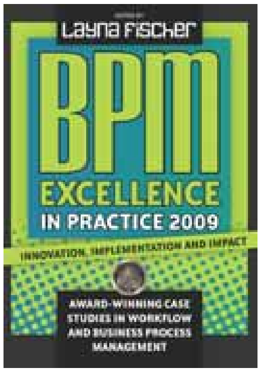
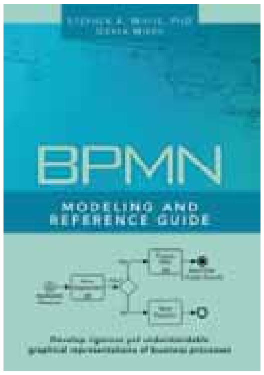
NEW E-BOOK SERIES ($9.97 EACH)

- Introduction to BPM and Workflow
- Academic
- Financial Services
- Government
- Healthcare
- Industry
- Transport
- Utilities

NON-PROFIT ASSOCIATIONS AND RELATED STANDARDS RESEARCH ONLINE

- AIIM (Association for Information and Image Management)
  http://www.aiim.org
- AIS Special Interest Group on Process Automation and Management (SIGPAM)
  http://www.sigpam.org
- BPM and Workflow online news, research, forums
  http://bpm.com
- BPM Research at Stevens Institute of Technology
  http://www.bpm-research.com
- Business Process Management Initiative
  http://www.bpmi.org *see* Object Management Group
- IEEE (Electrical and Electronics Engineers, Inc.)
  http://www.ieee.org
- Institute for Information Management (IIM)
  http://www.iim.org
- ISO (International Organization for Standardization)
  http://www.iso.ch
- Object Management Group
  http://www.omg.org
- Open Document Management Association
  http://nfocentrale.net/dmware
- Organization for the Advancement of Structured Information Standards
  http://www.oasis-open.org
- Society for Human Resource Management
  http://www.shrm.org
- Society for Information Management
  http://www.simnet.org
- Wesley J. Howe School of Technology Management
  http://howe.stevens.edu/research/research-centers/business-process-innovation
- Workflow And Reengineering International Association (WARIA)
  http://www.waria.com
- Workflow Management Coalition (WfMC)
  http://www.wfmc.org
- Workflow Portal
  http://www.e-workflow.org

MORE UNIQUE BOOKS ON BPM AND WORKFLOW FROM FUTURE STRATEGIES, PUBLISHERS (WWW.FUTSTRAT.COM)

	2008 BPM & WORKFLOW HANDBOOK ***Spotlight on Human-Centric BPM*** Human-centric business process management (BPM) has become the product and service differentiator. The topic now captures substantial mindshare and market share in the human-centric BPM space as leading vendors have strengthened their human-centric business processes. Our spotlight this year examines challenges in human-driven workflow and its integration across the enterprise. **Retail $95.00**
	2009 BPM & WORKFLOW HANDBOOK ***Spotlight on BPM in Government*** The question, “How can governments manage change organizationally and be agile operationally?” is answered in this special spotlight on BPM in Government with specific emphasis on the USA government where agencies, armed forces, states and cities are facing almost insurmountable challenges. **Retail $75.00**
	BPM EXCELLENCE IN PRACTICE 2009 ***Innovation, Implementation and Impact*** Award-winning Case Studies in Workflow and BPM These companies focused on excelling in *innovation*, *implementation* and *impact* when installing BPM and workflow technologies. They recognized that implementing innovative technology is useless unless the organization has a successful approach that delivers—and even surpasses—the anticipated benefits. **$49.95**
	BPMN MODELING AND REFERENCE GUIDE ***Stephen A. White, PhD, Derek Miers*** **Understanding and Using BPMN** Develop rigorous yet understandable graphical representations of business processes Business Process Modeling Notation (BPMN^{TM}) is a standard, graphical modeling representation for business processes. It provides an easy to use, flow-charting notation that is independent of the implementation environment. **Retail $39.95**