

Activity 4

Activity 2

The Ultimate Guide to BPMN2

Activity 3

The standard that bridges the needs
of IT and business for **business process management**

Contents

Why BPMN Matters 3

What is BPMN? 4

The ABC's of BPMN 5

A means for business & technical collaboration 6

The 4 categories of BPMN elements 7

BPMN elements in 4 categories 8

Workflow elements 9

Organizing elements 10

Readability elements 11

Special behavior elements 12

The 3 levels of BPMN complexity 13

BPMN at 3 levels of complexity 14

Basic BPMN 15

An example with basic BPMN elements 16

Intermediate BPMN 17

Intermediate activities 18

Intermediate sequence flow 19

Intermediate gateway 20

Intermediate events: catch and throw 21

Intermediate events: messages and signals 22

Intermediate events: timers and errors 23

Intermediate events in a process model 24

Summary 25

Sources and further reading 26

Why BPMN Matters

Business Process Model and Notation 2.0 (BPMN2) is one of the best things to happen in business process management in a long time.



Finally, both the business and technical sides of the organization can share a common language – something that they can both understand and that meets their respective needs for precision and flexibility. This shared language is empowering new ways of working together

- and it results in the deployment of new and more flexible applications.

At Bonitasoft, the leading provider of open source BPM solutions, we are

mindful of the power and potential of shared standards. BPMN 2.0 is a natural fit with what we do.



We realize that many people and organizations who could benefit from BPMN have yet to give it a try.

It may be that you've been putting it off under the mistaken assumption that you need to be an expert to use BPMN. Or it may be because the standard itself, and many of the things written about BPMN, are bit unwieldy and hard to dissect.

In fact, BPMN is much simpler than you might think; expertise comes over time – but starting is actually pretty easy.

And we believe the benefits can become quickly apparent. In fact, the nice thing about BPMN is that it is so structurally sound that once you master the Basic BPMN level elements, your knowledge and capability will improve quickly; you'll learn what you need from the intermediate BPMN level elements for extending the model, and the technical team will pick up the advanced BPMN level to complete the execution capability.

We offer this Ultimate Guide to help you to get familiar with the basics and give BPMN a try.

We are convinced you will find it powerful, adaptable and remarkably easy. Whether you are a business professional or a developer, BPMN2 is your path to better processes, improved management, and more efficiency.

Miguel Valdes Faura,
Bonitasoft CEO and founder

What is **BPMN**?

The ABC's of BPMN

If you've heard of BPMN but aren't really sure what it is or what it does, you are not alone.

But, before we talk about what BPMN is, let's talk about what it is not...

It is not a "system."

You can't "buy" a BPMN – it is a standard for business process collaboration and for IT development.

It is not just for business or just for IT– it is a shared, common language.

It is not only for experts.

If you are at all familiar with flow charting, you can dive in immediately.

BPMN = BPM + N

A **business process** model is a representation of an organization's processes. A model can be analyzed and improved.

Definitions

BPM Business Process Management

The discipline of managing processes as the means for improving business performance outcomes¹

BPMN Business Process Model and Notation

A graphical representation for specifying business processes in a business process model²

BPMS Business Process Management Suite

Application infrastructure to support BPM projects and programs... from process discovery, definition and design to implementation, monitoring and analysis, and through ongoing optimization¹

¹ Gartner Research

² Object Management Group

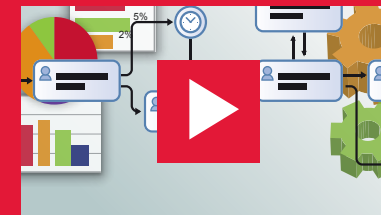
Notation consists of graphic symbols to represent action, flow, or behavior of a process.

In a BPMS, BPMN notation represents coding instructions that are executable.

BPMN provides a notation that can be readily understandable by all users:

- from the business analysts who model the processes conceptually,
- to the technical developers responsible for implementing the technology for the processes,
- to the people who will manage and monitor the processes.

See More



What is BPM?

A means for business & technical collaboration

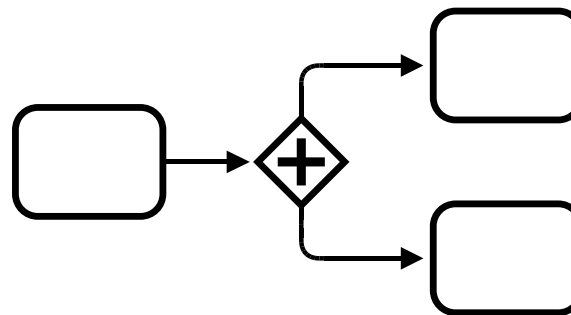
BPMN provides a way to quickly diagram business functions.

Use it to draw a process graphically.

The visual model will be translated quickly and easily into software that will run the process.

With BPMN, business people can define what they want, simply but with a high degree of precision; and IT professionals can communicate with each other and with business people about the model in a clear, common framework.

BPMN works for any kind of management, operation and support process. By developing a model with BPMN, you can collaboratively improve communications with decision makers about the nature and health of a process; you can collaboratively initiate improvements – and you can collaboratively move toward automating those improvements.



source: Business Process Model and Notation, Version 2, January 2011 by OMG

BPMN may look familiar

BPMN has been around for almost a decade and much in BPMN2 remains from the 1.0 version, especially the shapes and symbols.

One thing that has changed “behind the scenes” is the adoption of XML interchange format and the support BPMN 2.0 provides for turning a model and its notation into an executable process.

Open source and proprietary BPM vendors now have the capacity to take BPMN 2.0 input and turn it into process automation.

BPMN is not an execution language.

It is designed to be “consumed” by process engines and made into executable processes.

The 4 categories of BPMN elements

BPMN elements in 4 categories

The BPMN2 spec is long, dense and relatively complex

We can approach it by organizing BPMN elements into a few general categories.

With just a few elements from first three categories you can draw a business process diagram and begin to build and understand a process.

Let's look more closely at what they represent.

In the **BPMN2** spec

98 visual elements

508 pages

300 figures

313 tables

3 annexes

13 collaborating groups

Workflow elements	Organizing elements	Readability elements	Special behavior elements
Activities Events Gateways Sequence flow	Pools Swimlanes Groups	Annotation Links	Messages/message flow Signals Timers Errors Repeating Correlation

Workflow elements

Workflow elements include *activities, gateways, events, and the sequence flow* that connects them.

Each of these elements have several types, and all of these types can be connected in a sequence.

Activities

Tasks that are performed in the process- by humans, by automation, or by subprocesses



Events

Used to start or end a process, and to manage specific actions during a workflow



Event

Gateways

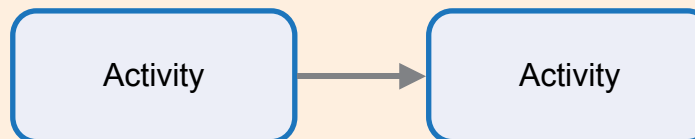
Used to separate or join process flow



Gateway

Sequence flow

Used to show how the workflow moves



See More



Getting Started with BPM

Organizing elements

Organizing elements include *pools* and *swimlanes*. Think of these as the container for the process flow.

Pool

Contains a single, complete process. Workflow cannot leave a pool - we need to transfer action or data from one pool/process to another using events.

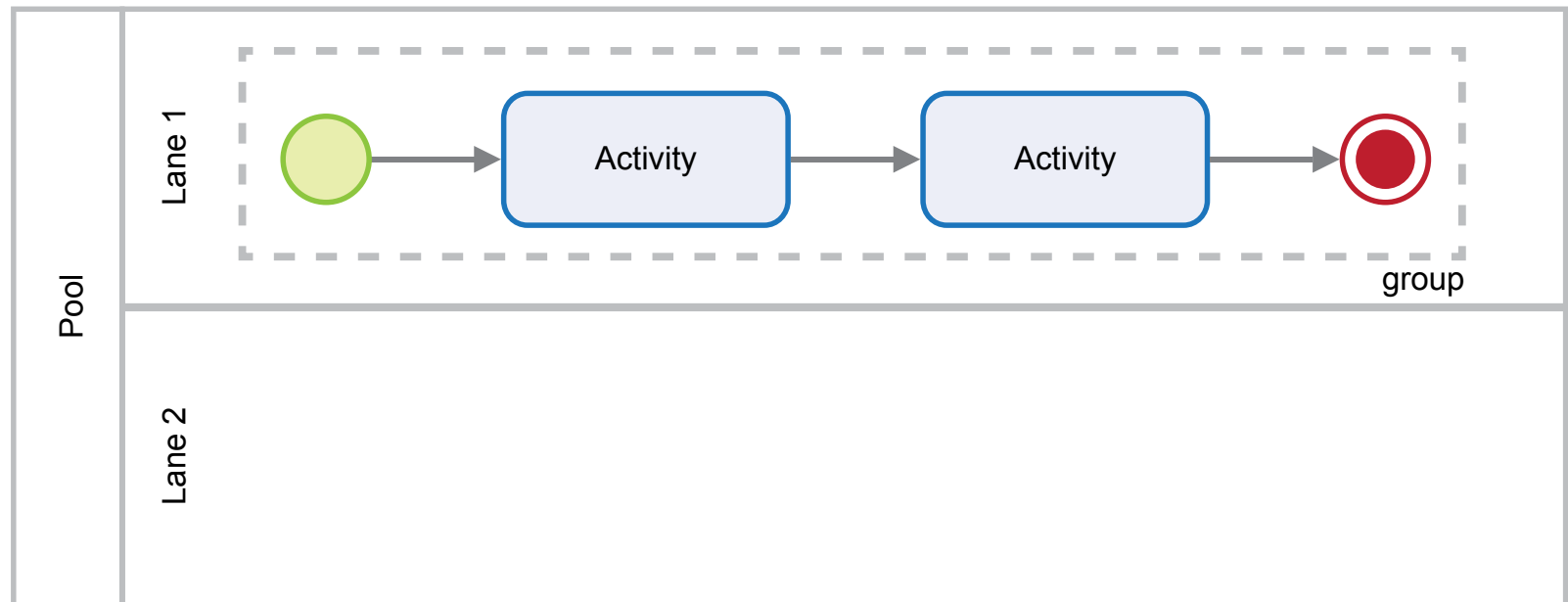
Swimlane

Used to help organize the process based on who does what. In a lap pool, swimlanes keep the swimmers from crashing into one another.

Workflow crosses swimlane boundaries as if they did not exist – they are purely for organizational clarity.

Group

Used to enclose a grouping of graphical elements. It does not affect sequence flow.



Readability elements

Readability elements include *annotations* and *links*.

These elements help make a model readable. They have no effect at all on the actual process flow.

Text annotation

Allow you to paste notes all over a model with explanations for clarity (a great tool for beginning modelers!)



Annotation

Links

Allow you to cut a process that has become too long to read easily, and simply continue the process on another line.



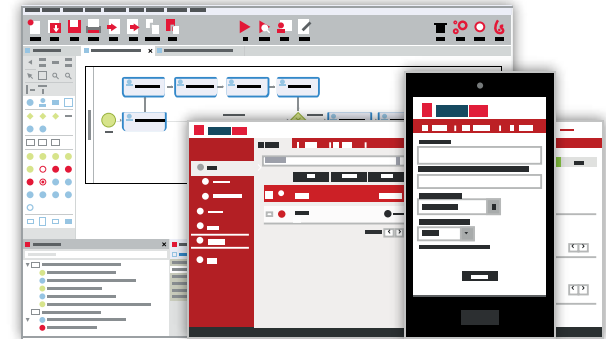
Throw link



Catch link

BonitaBPM

Turn diagrams into executable apps with this powerful BPMN2 process modeler



and it's
Open Source

[Download](#)

See More

Bonita BPM documentation:
Process Modeling

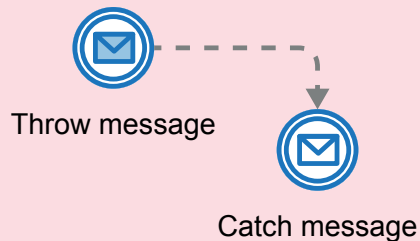
Special behavior elements

Special behavior elements include a specific set of *events*, *repeating*, and *correlation*.

These elements allow us to design executable workflow that can behave in complex ways.

Messages and message flow

Used to transfer action or data from one pool/process to another and to correlate related processes



Signals

Used to send data to multiple activities simultaneously



Throw signal Catch signal

Timers

Used to launch periodic activities, or to ensure that an activity happens within a specified deadline



Timer

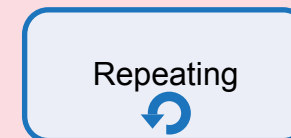
Errors

Used to define behavior when the system encounters error



Repeating

Used to repeat behavior, such as multiple launches of the same task, or repeating the same task multiple times



The 3 levels of **BPMN** complexity

BPMN at 3 levels of complexity

BPMN symbols serve a dual purpose.

They visually represent a process flow. They translate to executable code that allows a visual process model to be executed as an application.

Recall that we can organize BPMN modeling elements into a few general categories:

- Workflow elements
- Organizing elements
- Readability Elements
- Special behavior elements

Note that Basic BPMN is predominately **visual**. Intermediate and Advanced BPMN becomes **executable**.

Let's look at these BPMN elements at the three levels of complexity: **Basic, Intermediate and Advanced**

	Basic	Intermediate	Advanced
Activities	Abstract task	Human task Service task Call activity	Event subprocess
Events	start end	In-line and boundary events - See special behavior	
Gateways	XOR AND	Inclusive	
Sequence flow	Sequence flow	Conditional flow Default flow	
Special behavior		Message Timer Error Signal	Looping Multi-instantiation Transaction Compensation Correlation

Basic BPMN

Basic BPMN is useful for modeling when details have not been worked out.

Activities, events, gateways, and sequence flow all have Basic BPMN level versions.

Basic activities are abstract, or undefined. **Basic events** include **start** and **end** events. A **start** begins a process and an **end** terminates it.

Start = go

End = stop

Basic gateways

AND (also known as parallel)

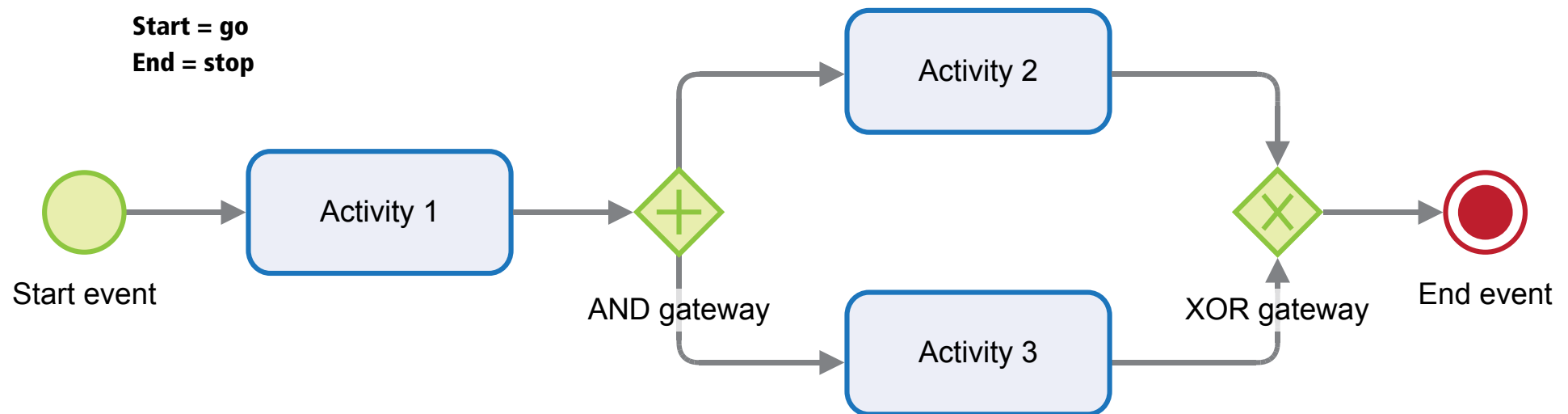
All inputs must be received (in any order) before the process can continue. All outputs are activated – process continues in parallel.

XOR (also known as exclusive)

Only one input is needed for the process to continue.

Only one output is followed – a condition is needed to determine which one.

Basic sequence flow simply directs process flow from activity to activity.



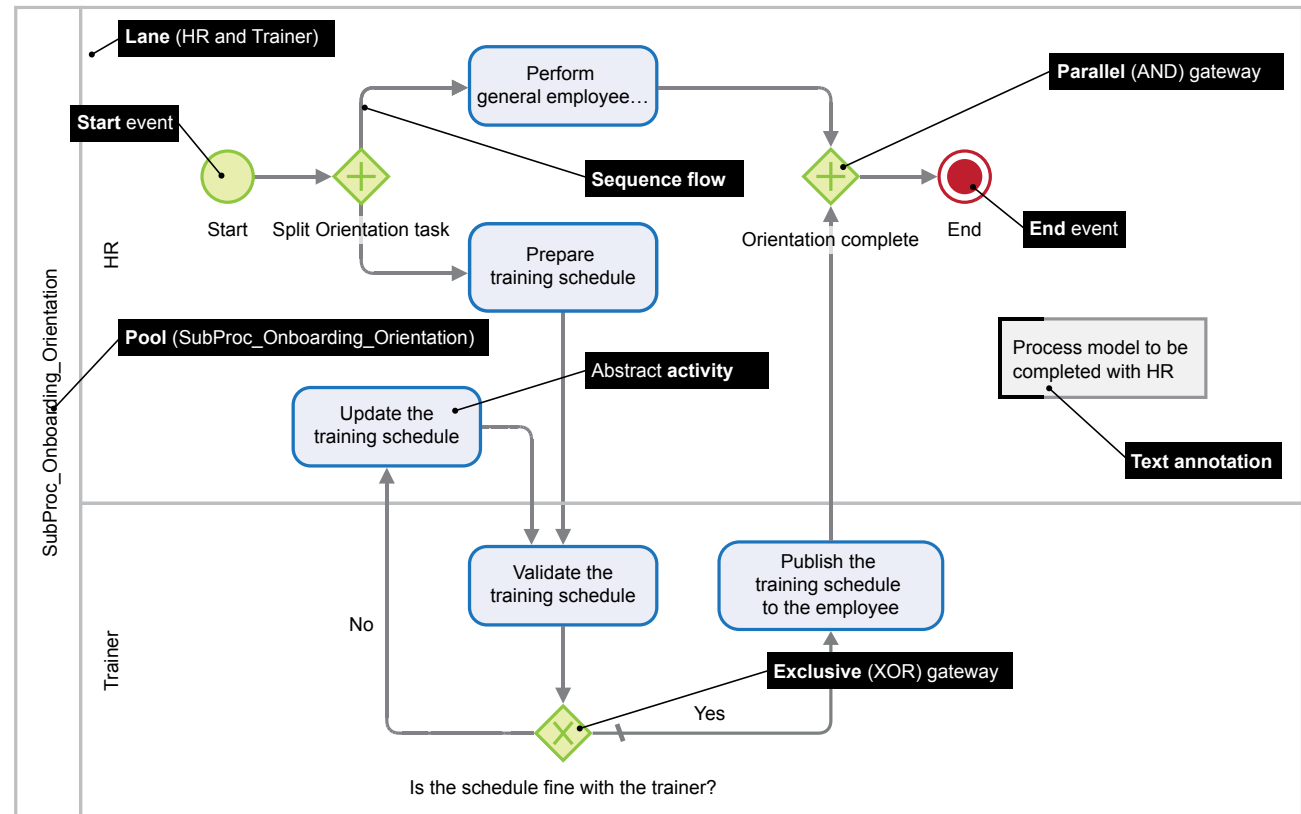
An example with basic BPMN elements

Start with the basics: *abstract activity, start and stop events, gateways, and sequence flow.*

For example, a generic new employee orientation and training process modeled in basic BPMN elements looks like this.

Imagine a token being moved through the diagram – like a traditional board game. This can help clarify how the features of the model control the movement of the token as you add complexity. When a start event is triggered, a new “instance” of a process begins. Think through what happens to a single token traversing a single pathway at a time.

BPMN 2.0, Thomas Allweyer



New employee orientation and training process

Intermediate BPMN

To make a visual model executable, begin to apply intermediate BPMN.

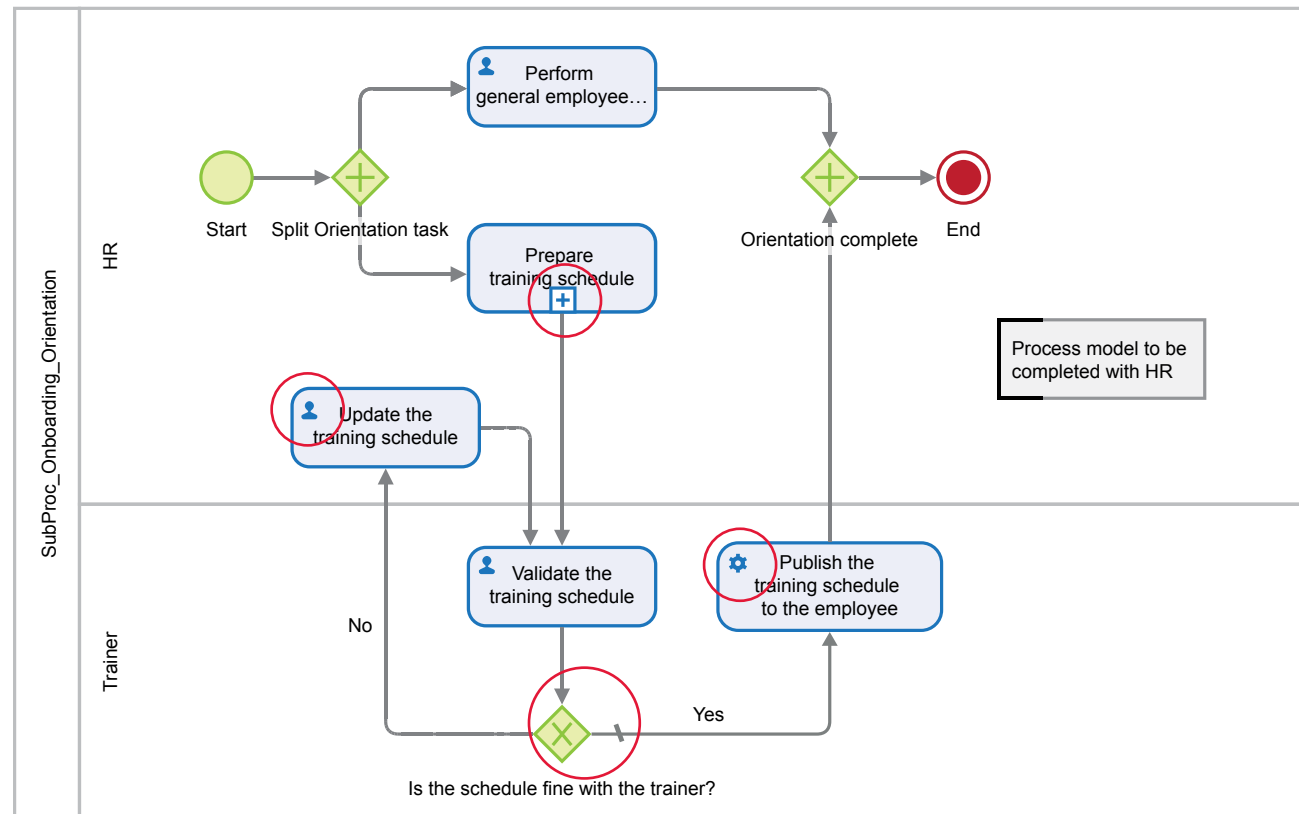
In an executable process, the flow model becomes an actual process application!

As you advance with BPMN, begin making your BPMN “executable” – to ultimately turn it into an automated process.

BPMN 2.0 is not just a notation. Implemented through a BPMN modeling tool, it provides programming instruction that a process engine uses to execute the process.

The previous example is a simple model that clearly shows visually what happens in the process.

The example on this page and the next shows how the model is extended as you begin to apply intermediate BPMN.






New employee orientation and training process

Intermediate activities

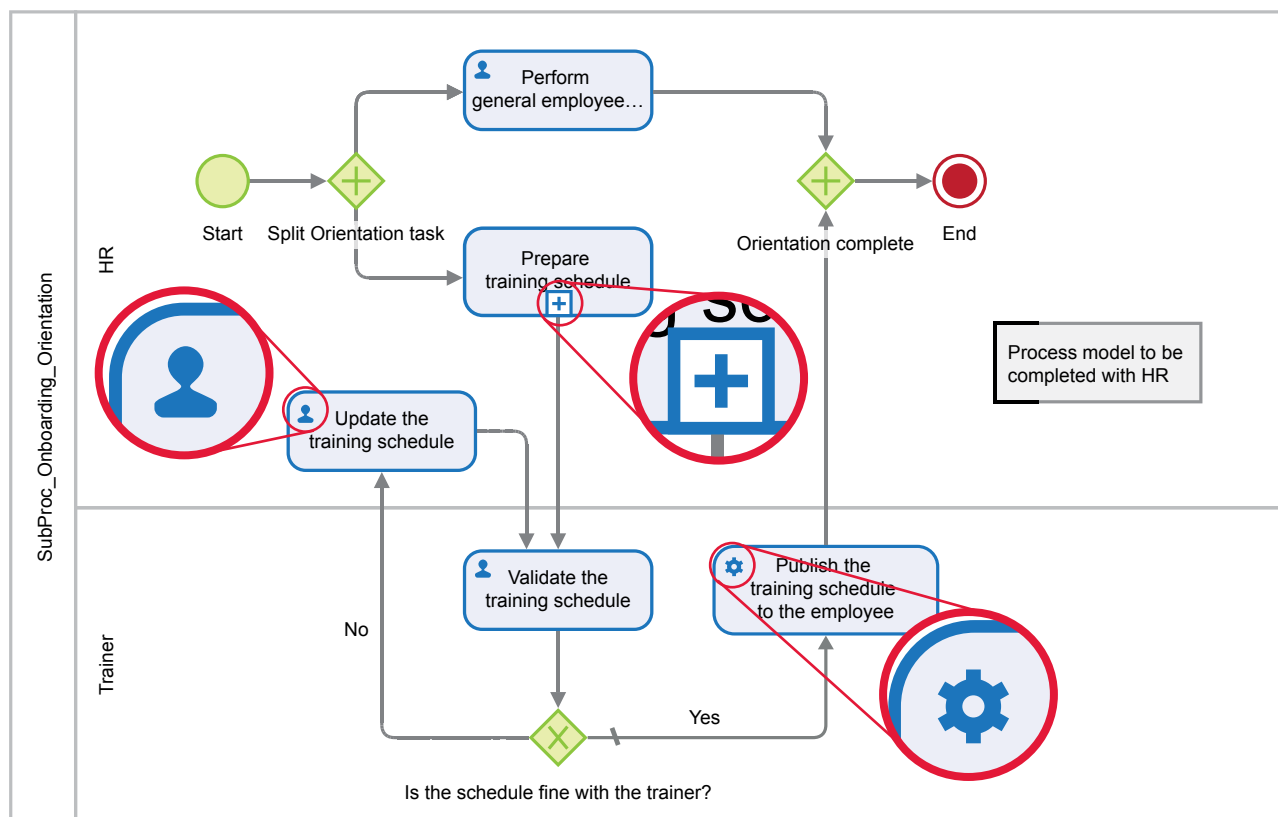
Intermediate *activities* include *human*, *service*, and *call* activity

Activities need to be differentiated – is each task performed by a person or is it automated or performed by the software?

Or, is it a subprocess in its own right?

-  **Human activity** is a step that must be done by a person
-  **Service activity** is an automated step
-  **Call activity** represents a subprocess

“Prepare training schedule” is a **call activity**. It is linked to a subprocess (a “child” of the original parent process). At this point in the process, the “token” is passed to the subprocess, and when it has completed its passage, it is passed back to the parent process. This is a super-useful aspect of BPMN.



Using this notation, you can model a top-level parent process that can be quite simple. It can call a series of subprocesses that are entirely independent workflows.

This means they can be modeled independently and modified as needed without necessarily changing the parent process.

Intermediate sequence flow

Intermediate sequence flow includes *conditional* and *default* flows.

Sequence flow in intermediate BPMN needs to be defined as conditional or default, so the "flow token" knows which path to follow.

Basic sequence flow is simply automatic (as soon as an activity is completed, the process moves to the next task in the sequence).

Conditional sequence flow

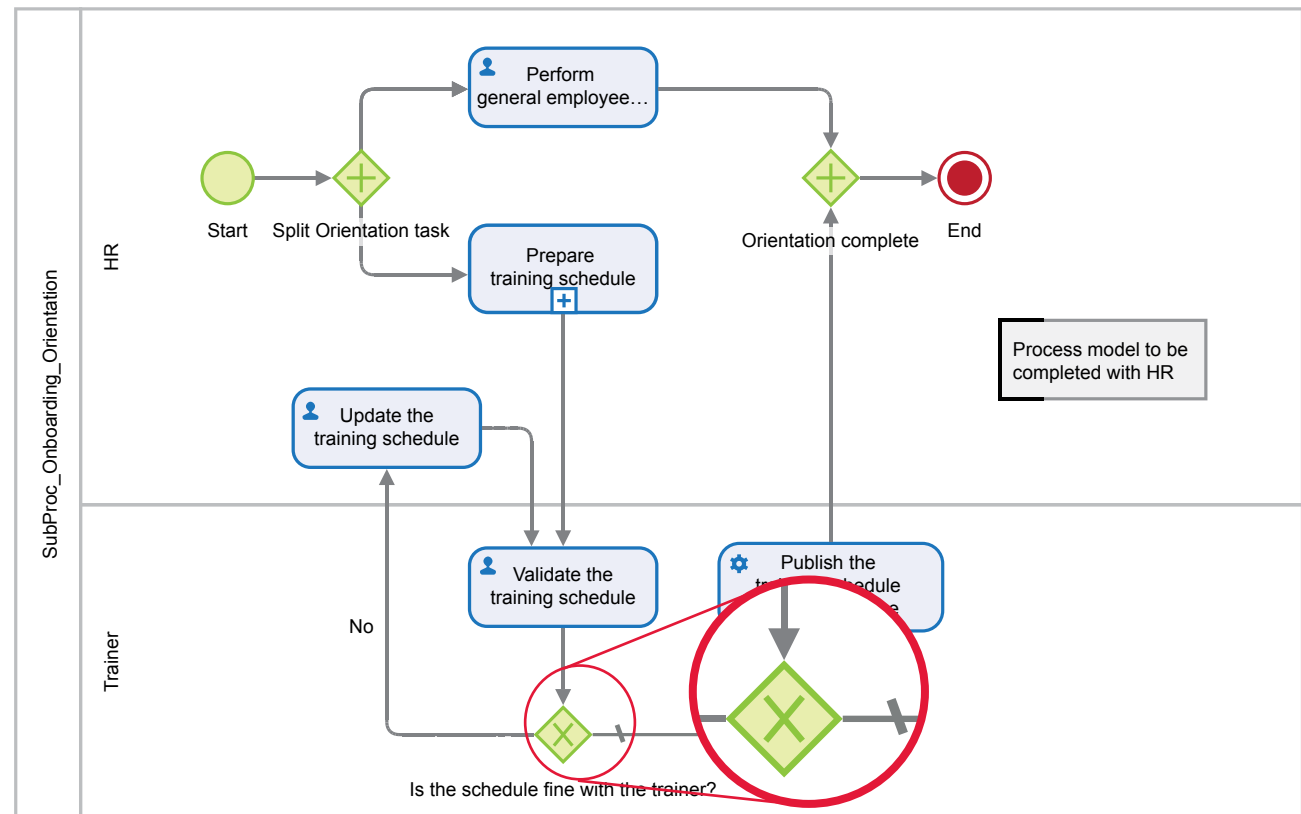
Some specified condition(s) must be met so the process can "choose" the next task from among two or more options.

Conditional flow is what it sounds like: an IF-THEN condition is defined. In this (Boolean) example:

If the schedule is ok with the trainer, this condition = true.

If the schedule is NOT ok with the trainer, this condition = false.

Default flow allows you to direct



flow if, for some reason, no conditions are met. The flow token always has a direction to take, even if there is a data entry error somewhere that might invalidate the defined IF-THEN condition.

Default flow is marked with a \

Sequence flow can't cross a pool boundary. To communicate flow between pools (processes), use *messages*.

Intermediate gateway

The intermediate gateway *inclusive* offers finer control of process flow

Outputs from inclusive gateway

The inclusive gateway can fire **multiple outputs simultaneously**.

It requires **conditions** on the outgoing sequence flows.

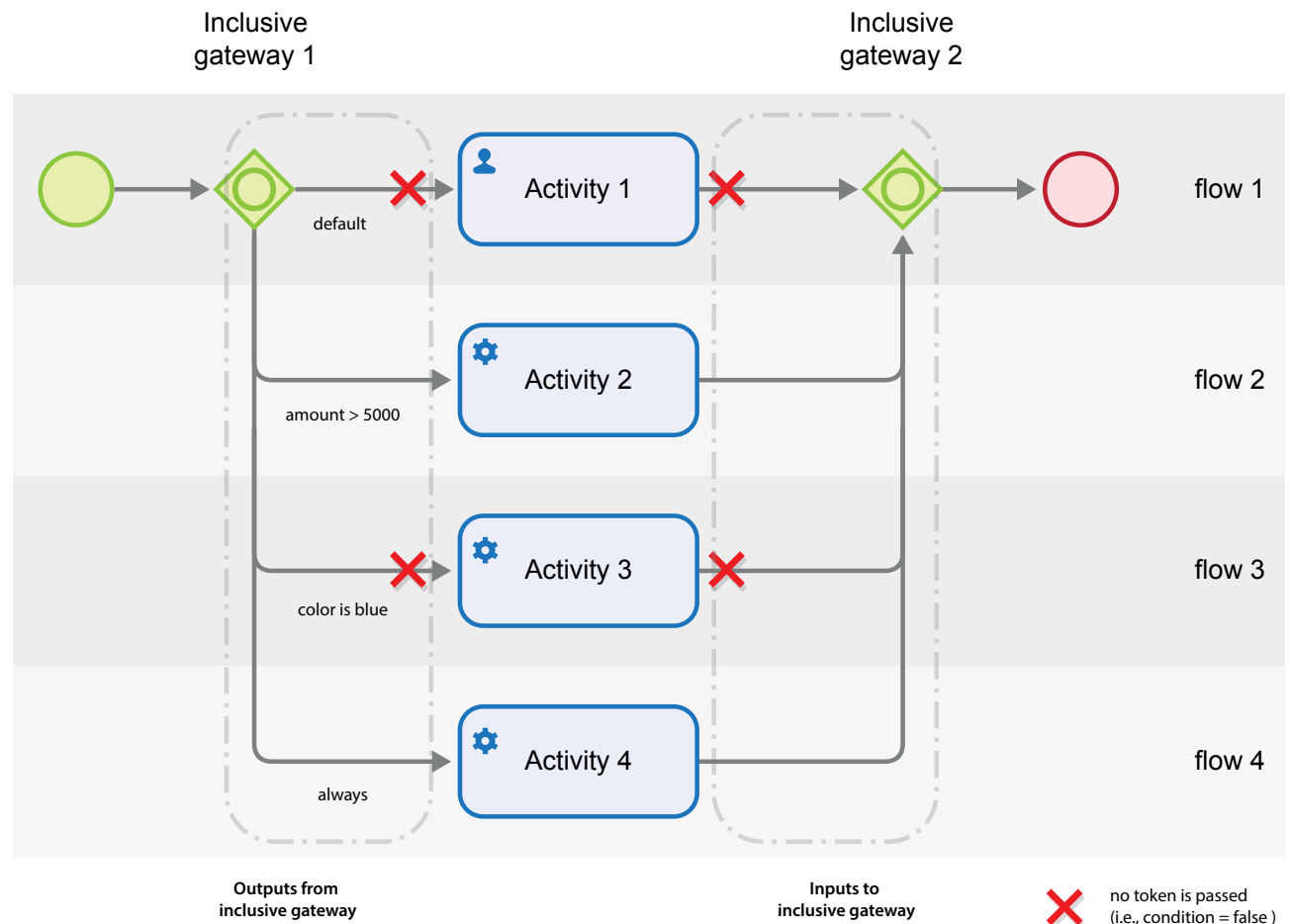
Example

Condition	Value
amount	5000
color	red

In this example, flows 2 and 4 **meet the flow condition**. Flows 1 and 3 do not – so no token passes.

Inputs to inclusive gateway

The inclusive gateway **waits for all incoming inputs (tokens)**. All valid inputs must be received before the process flow can continue. The engine recognizes which inputs it must wait for (i.e., flows 2 and 4).



Intermediate events: catch and throw

Intermediate events are either *throw* or *catch* events.

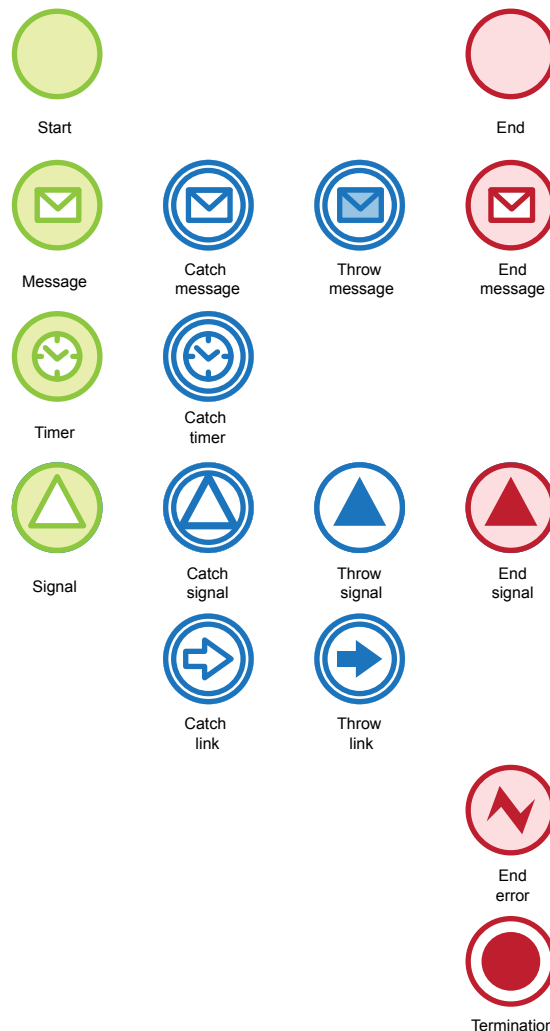
Mastery of special start, end, and in-flow “intermediate” events is key to mastery of intermediate BPMN.

BPMN events are defined generally as “throw” (think of these as senders) and “catch” (think of these as receivers.)

Mix-n-match events

Events can have multiple characteristics.

- **solid** - throws or sends events
- **empty** - catches or receives events
- **green** starts a process
- **red** ends a process
- **blue** intermediate, takes place within the flow of a process



A catch event can be located anywhere along a process flow. The BPMN spec somewhat confusingly refers to this case as an “intermediate event.” If you stick to thinking of events as throw/send and catch/receive, BPMN may be easier to understand.

Special start events (message, timer, signal, error) allow you to trigger processes without direct human interaction, as they are set to “catch” information send from elsewhere.

“Elsewhere” means from a throw event somewhere in another process, and this can be an end event.

In this case, the end of one process can trigger the start of another process.

Intermediate events: messages and signals

Messages and signals carry data across pool boundaries.

Messages broadcast narrowly to single receivers, while signals broadcast widely to many receivers.

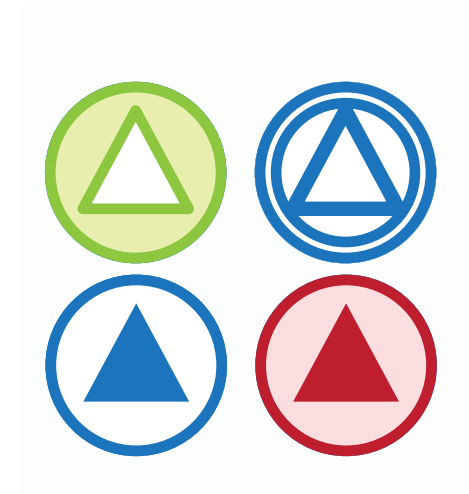


Message

You can start a process with a message. In BPMN, message is specifically defined as the means by which data can be transferred between processes. In fact, it's the only way.

With BPMN you can start a process with data received from a different process.

And conversely, if you want to send data to another process, use an intermediate send message (anywhere in the process flow) or an end message.



Signal

Like messages, timers, and errors, signals can be caught from elsewhere and can start a process. A single "throw" signal is broadcast widely and can be received by multiple catch signals. This is useful when you want multiple actions to be triggered.

Intermediate events: timers and errors

Timers and errors can delay or pause a process, or send it on an exception path.

Like other intermediate events, timers and errors can start or end a process - or impose an action within the process flow.



Timer

Timers can be set to “go off” at specific intervals, or specific calendar-linked dates and times. For example, a start timer can go off every 24 hours, or on the first Tuesday of each month.

If the timer is a start event, the process starts when the timer goes off. If the timer is located in the process flow, the process waits until the timer goes off – and then it continues.

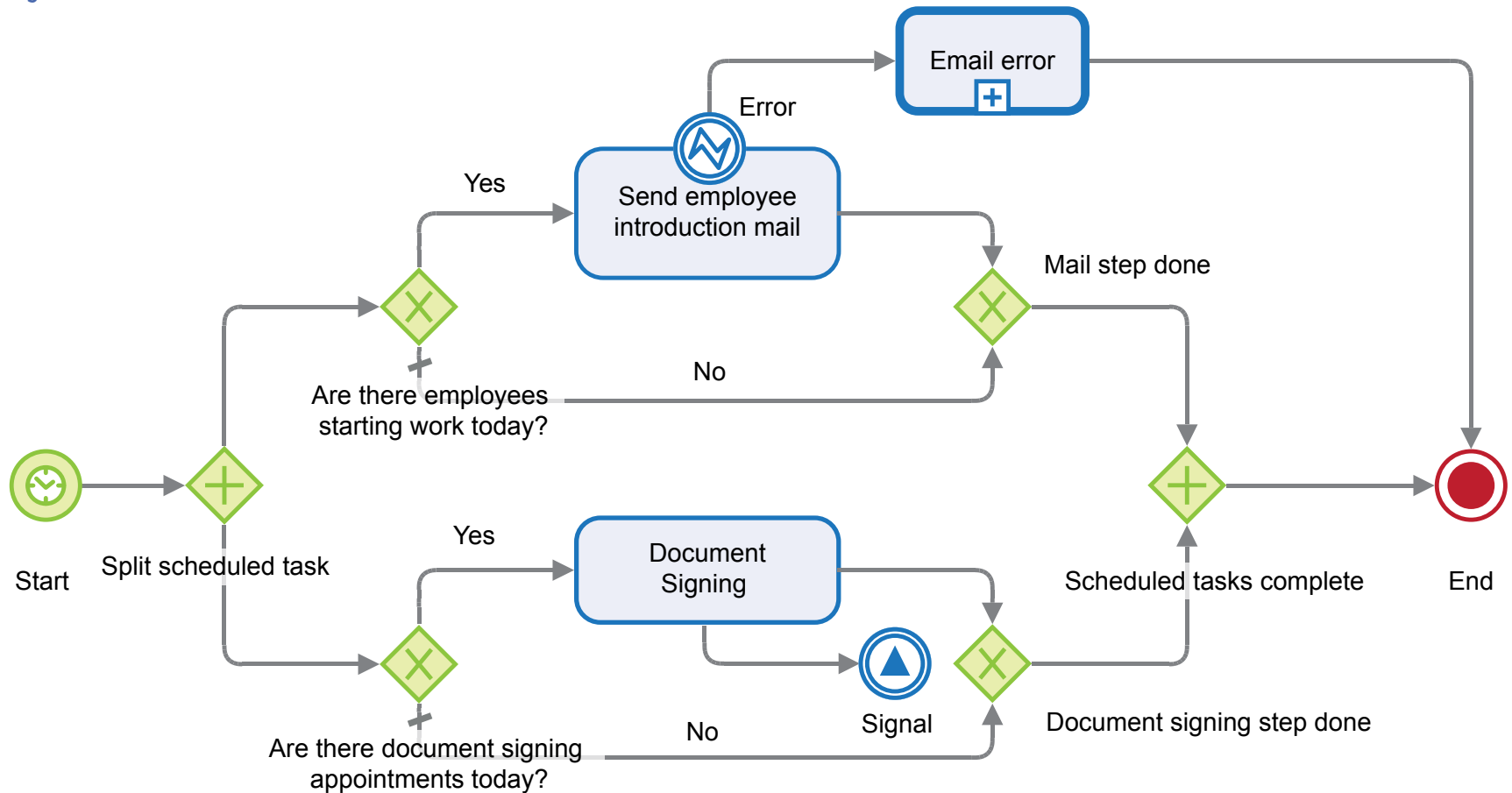


Error

Like messages and timers, errors can be caught - and can start a process, or a special error path within the sub-process.

Intermediate events in a process model

Messages, signals, timers, and errors specify workflow behavior.



Summary





With just 4 *categories* of basic and intermediate BPMN you can begin to build a deployable, executable process application

BPMN is a standard that allows business and IT to share a common language, which makes development of BPM applications for business by IT easier and more efficient.

BPMN is both a set of visual modeling elements, and a set of semantics for executable code represented by those elements.

Many of the visual elements in BPMN are similar to standard flow chart elements. Modeling with and interpreting models with BPMN is relatively straightforward.

BPMN elements can be categorized:

-  Workflow elements
-  Organizing elements
-  Readability elements
-  Special behavior elements

There are **Basic, Intermediate, and Advanced** elements in each of these categories.

Basic BPMN is useful for modeling.

Intermediate BPMN begins to make a model executable.

Advanced BPMN fully defines process behavior.

If you're designing a BPM software suite, read the BPMN2 spec...

If you're designing process applications, The Ultimate Guide is what you really need!

See More



VIDEO Create and Run Your First Process

Sources and further reading

BPMN Method and Style, 2nd ed, Bruce Silver, October 2011

OMG Business Process Model and Notation (BPMN) Version 2.0, January 2011

BPMN 2.0: Introduction to the Standard for Business Process Modeling, Thomas Allweyer, February 2010

See More



Introduction to BPMN Object Management Group

