Composition and Orchestration of Services

Oxford University

Software Engineering Programme

Sep 2015

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Business Process Management

- Hammer & Champy [1993] "A collection of activities that takes one or more kinds of input and creates an output that is of value to the customer."
- Davenport [1992] "A structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what."

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

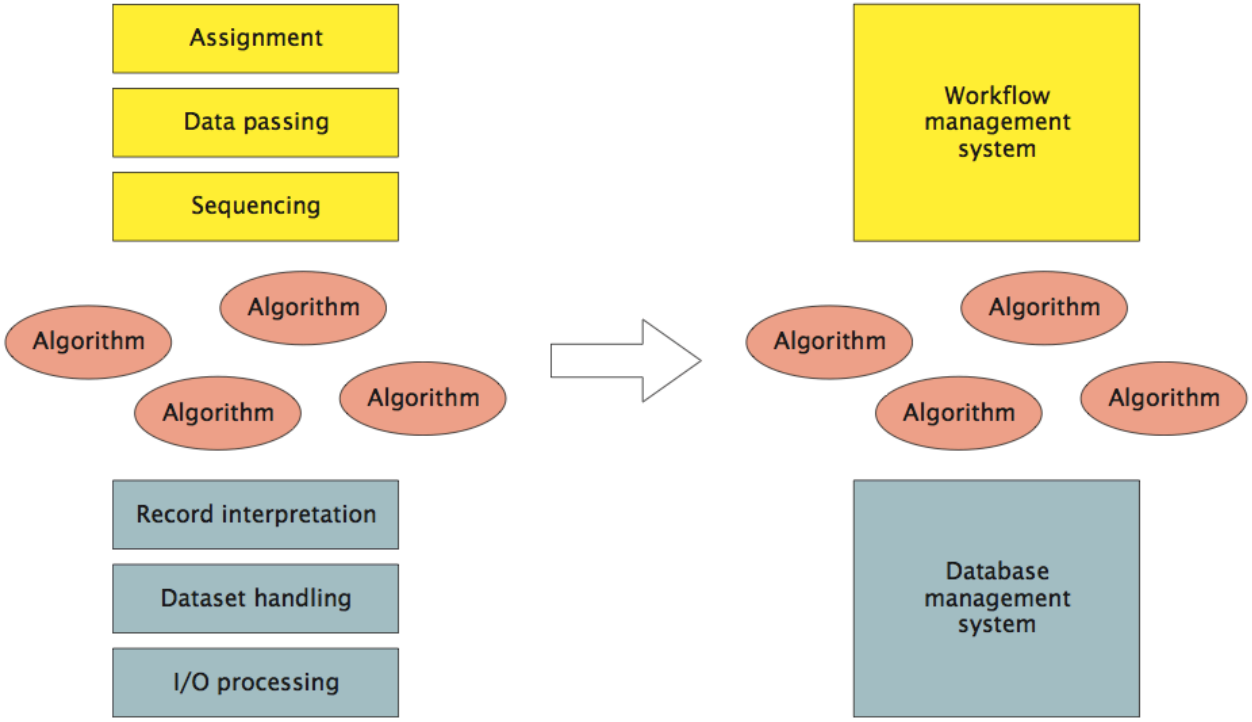
Composition

- Services provide *platform- and language-independent access* to *software components*
- But these components are *isolated*: they need to be *assembled* into *service-oriented architectures*
- Ideally, they should be recursively *composable* to form composite services in their own right
- *Workflow* languages for scripting or 'glue' between individual services
- BPMN, WSCI, WSFL, XLANG, BPEL. . .
- beyond mere *business protocol specifications* like RosettaNet, which are essentially paper specifications so can't be automated and won't scale

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Removal of Dependencies (Leymann and Roller)

- DBMS provides independence from data *representation*; workflow provides independence from control or data *flow*.



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Heritage

- E*nterprise application integration* (EAI)
  - resolving heterogeneity, typically via asynchronous *message brokers*
- *Workflow management systems* (WfMS): automating interactions
  - origins in *office automation*: admin processes
- P*roduction workflows*: from information between people to integration of systems
  - often associated with *business process re-engineering*: assessment, analysis, modelling, definition, implementation
- Service composition = EAI + WfMS

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

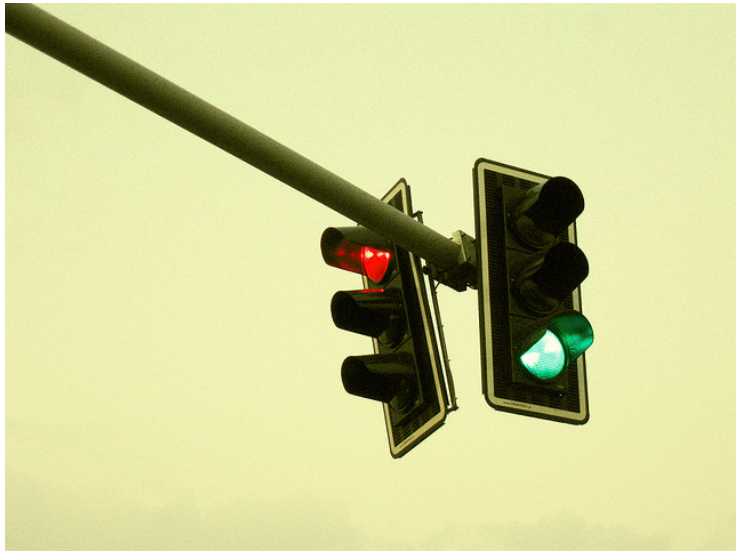
Motivations

- Model Business Processes
  - Understand what happens?
  - Who is responsible?
  - What is involved?
- Simulate
  - Improve and model
- Execute
  - Automate processes
  - Improve them more quickly
- Monitor
  - Get a real-time health status of processes

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Orchestration vs Choreography

http://www.flickr.com/photos/herrolm/



http://www.flickr.com/photos/tasuki/

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Orchestration vs Choreography

- *Orchestration*
  - Describes procedure
  - instructs participants globally – imperative; centralized
  - typically deterministic: ‘must’
- *Choreography*
  - Describes protocol
  - Constraints on interaction, but participants act locally – declarative; no ‘current state’
  - Usually non-deterministic: ‘may’
- Orchestra has a conductor, Ballet does not

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

WS-Choreography Description Language

- http://www.w3.org/TR/ws-cdl-10/
- Never got past Candidate Recommendation
- Captures the flow of messages between parties
- Temporal and logical dependencies between messages
- features sequencing rules, correlation, exception handling and transactions
- Not executable

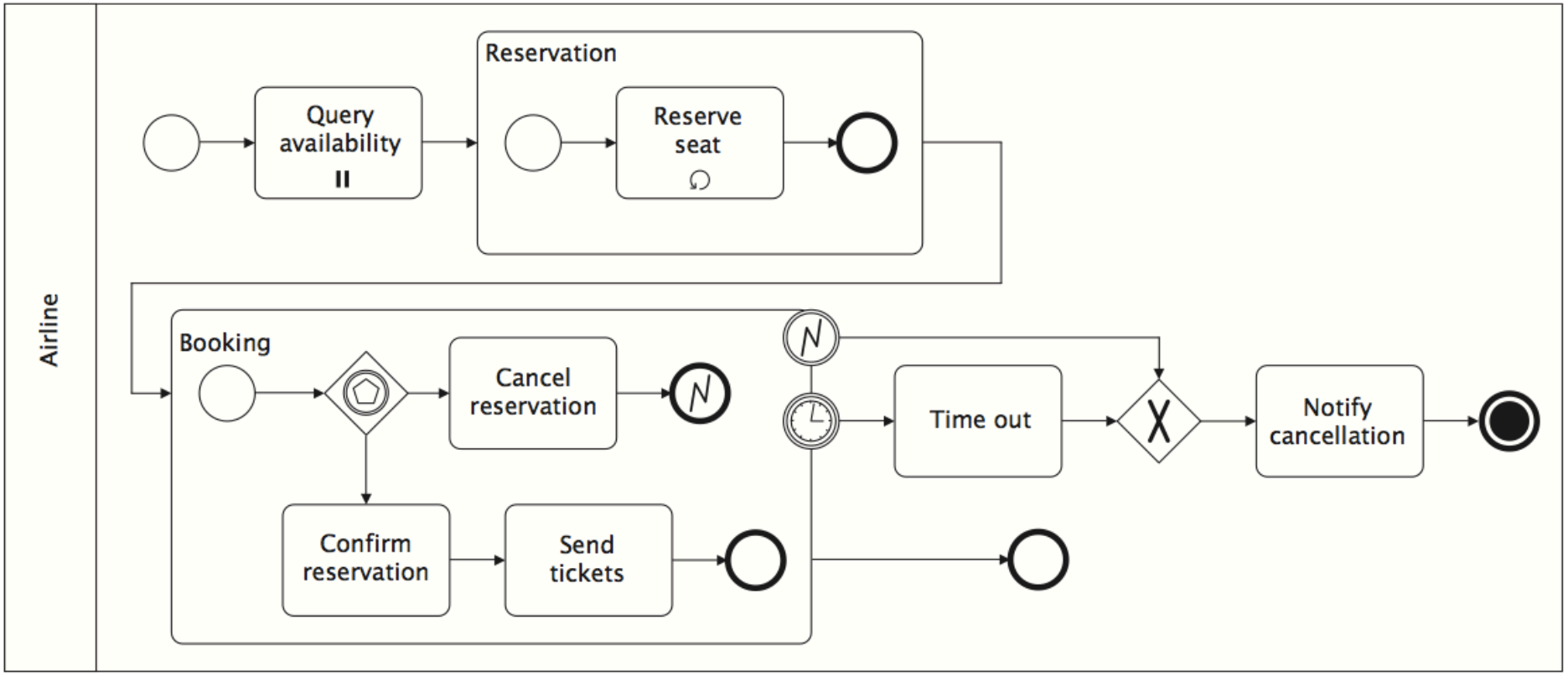
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

BPMN 1.1

- Designed to allow process designers to communicate
  - Think UML
- Activities, Gateways, Events
- Control and Data Flow
- Organization modelling (Pools, Swimlanes)

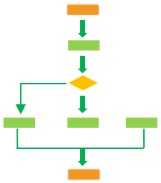
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

BPMN Example



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Amazon Simple Workflow Service



DECIDER

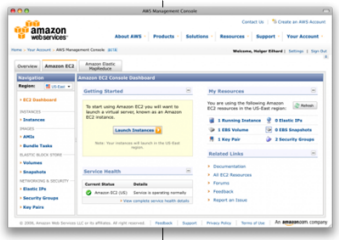
The decider implements the application's business logic

Gets Decision Tasks

Return Decisions

AWS

Amazon SWF



- Maintains distributed application state
- Tracks workflow executions
- Ensures consistency of execution history
- Provides visibility into executions
- Holds and dispatches tasks
- Provides control over task distribution
- Retains workflow execution history

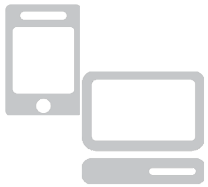
Get Activity Tasks / Return Results

Cloud

Workers for Activity 1

Get Activity Tasks / Return Results

Mobile



Workers for Activity 2

Get Activity Tasks / Return Results

On Premises

Workers for Activity 3

http://aws.amazon.com/swf/

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Business Process Execution Language (BPEL)

- Standardised XML language for executable processes
- Well defined execution
  - No deadlocks
  - Graphs must be acyclic
- Tied to WSDL concepts
- No built in support for human activities (though this has been added)
- No graphical notation

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

The main strength of BPEL
(IMO)

- BPEL is a completely executable standalone language
  - PartnerLinks define places where you can call WSDL services
  - Or where other parties can call WSDL Services into the process
- Deployment descriptor + BPEL can be executed without any Java or other language

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

The main weaknesses of BPEL
(IMO)

- Too much like a programming language
  - Need WS-HumanTask, BPEL4People and script or Java extensions to make it useful for real processes
- No swimlanes (explained in a minute)
- No common visual notation

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

BPMN + BPEL

- In theory:
  - Process experts design and model in BPMN
  - Developers/Implementors implement in BPEL
- No standard bridging/mapping
  - Double the effort

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

BPMN 2.0

- A notation for a subset of BPEL
- Execution semantics for BPMN
- Notational support for choreography
- The best of both worlds?

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

CMMN

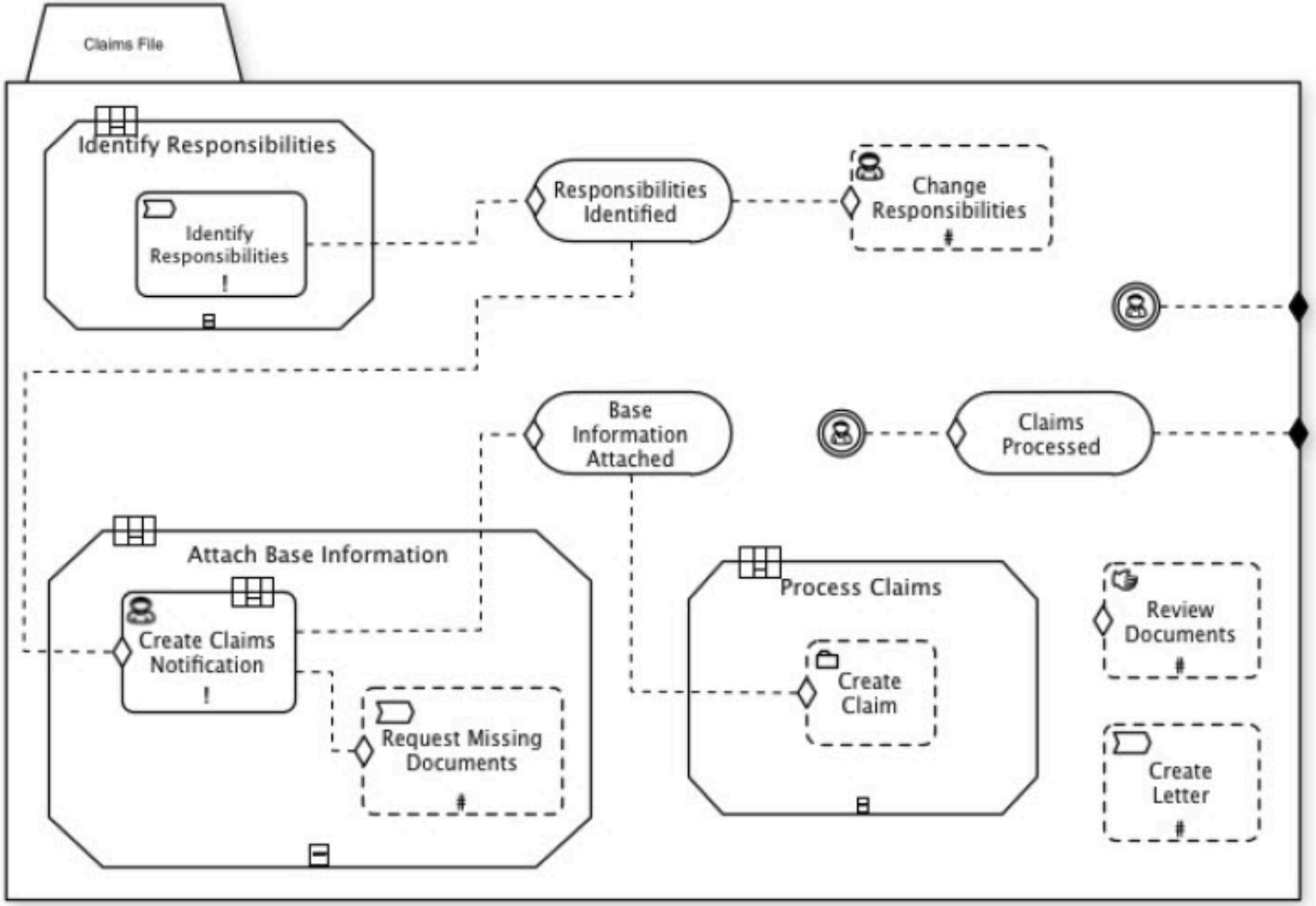
Case Management and Modelling Notation

- A specification from OMG for modelling how to handle cases
- A more flexible approach to workflow that BPMN or BPEL
  - Certain workflows are very clear
    - Building a car
  - Others are more flexible
    - Hand building a mandolin
- Imperative vs “Causative”

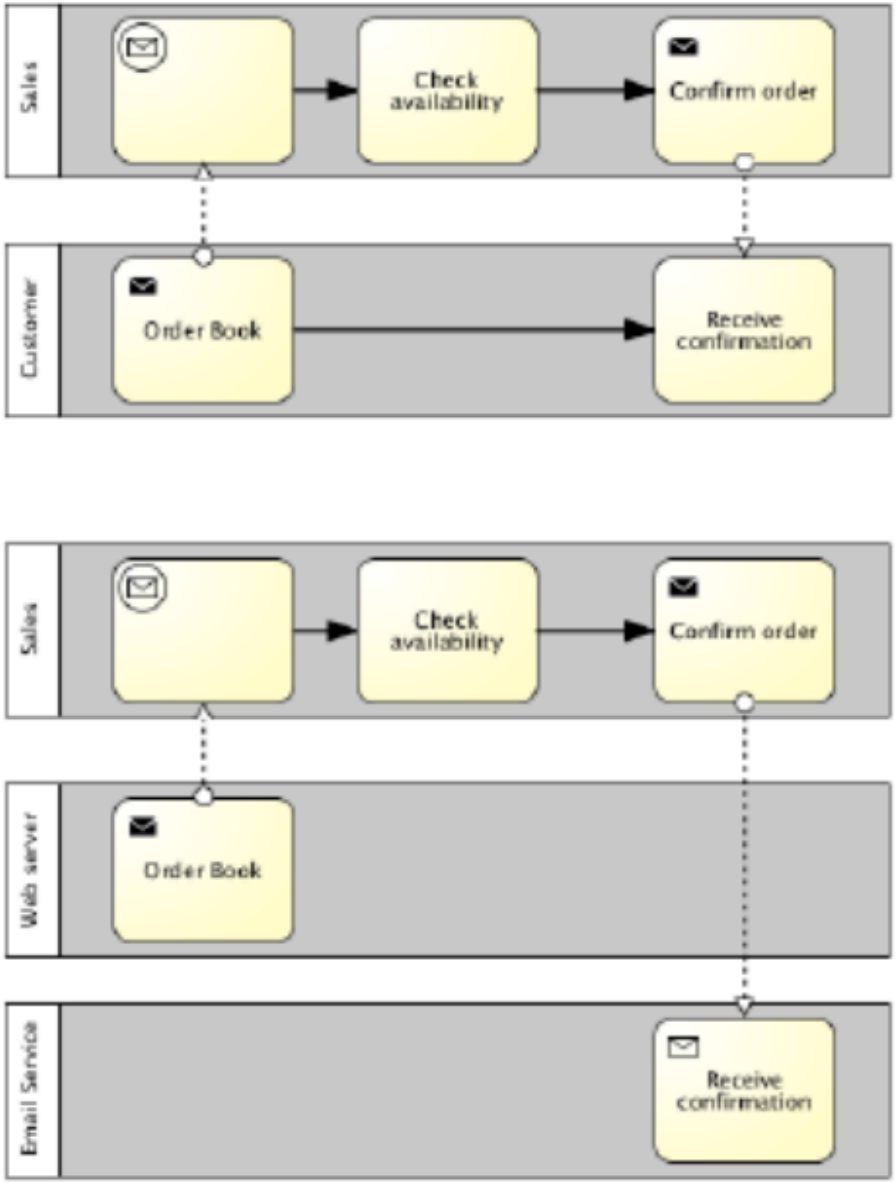
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

CMMN example

Source: http://brsilver.com/bpmn-cmmn-compared/

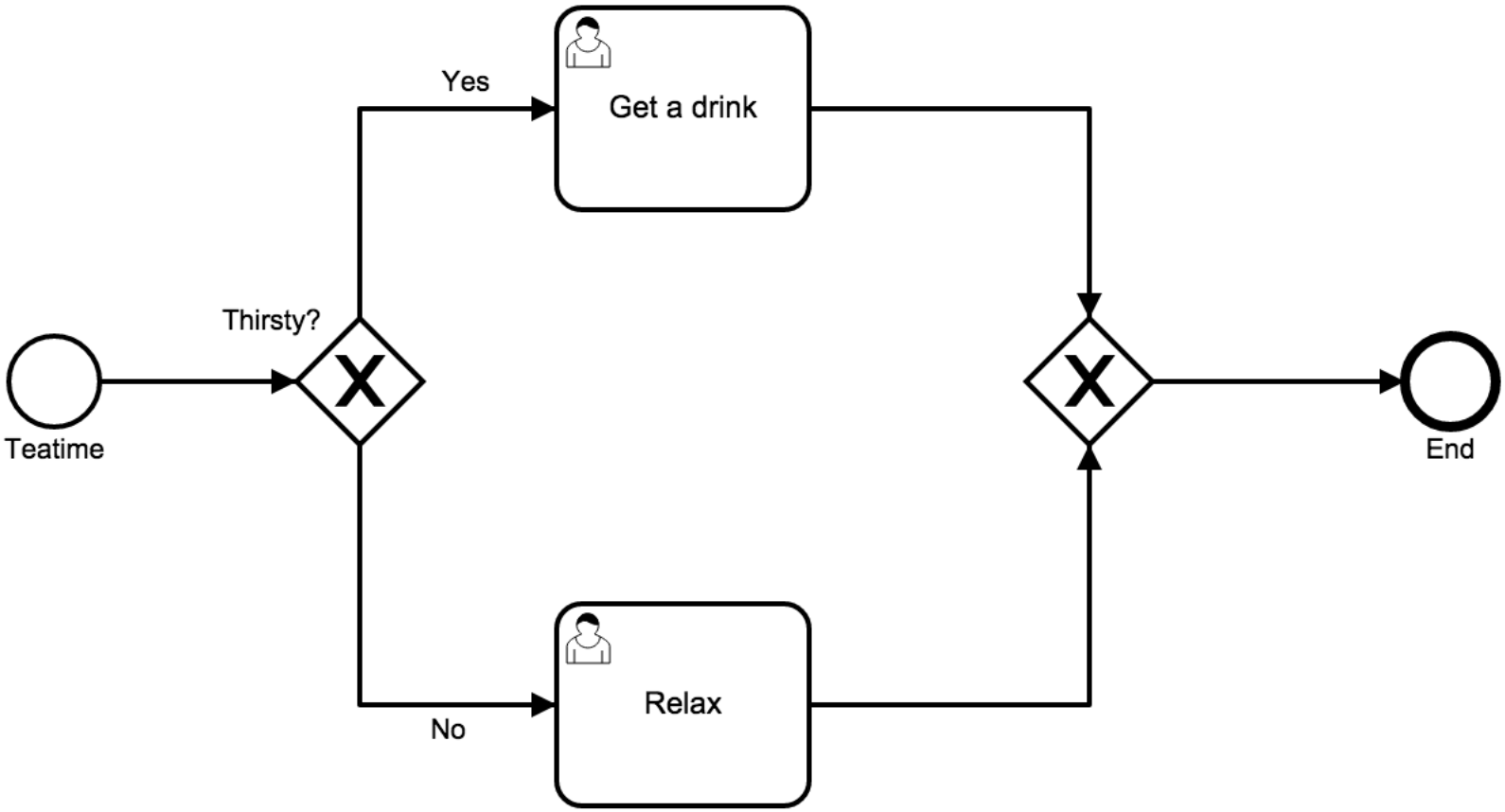


BPMN 2.0



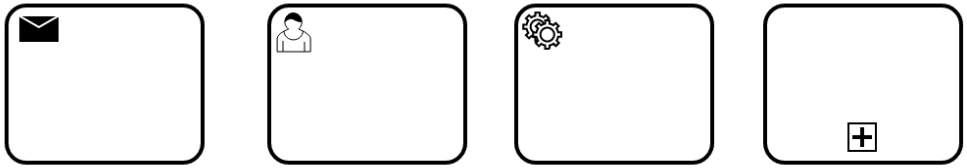
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

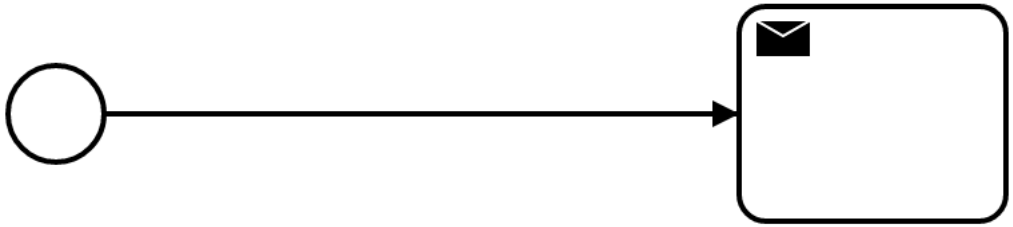
BPMN 2.0 Basics



BPMN Basic Constructs

- Events
- Activities
- Gateways
- Sequence Flows

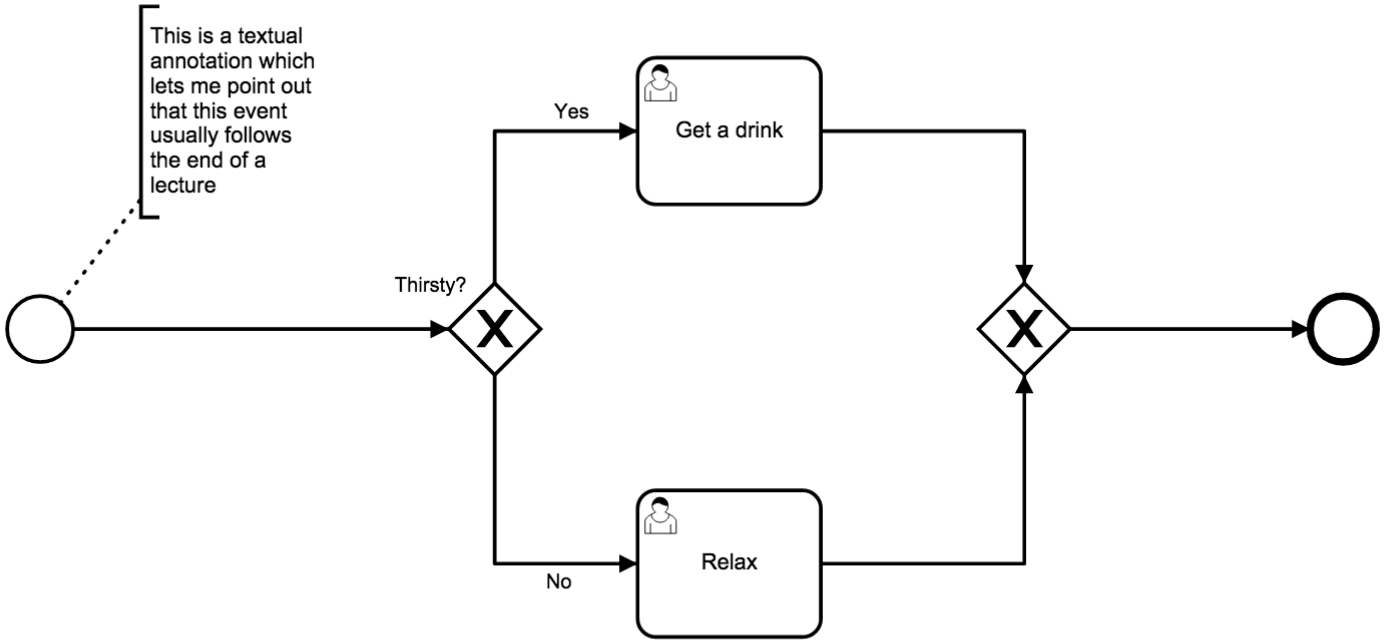




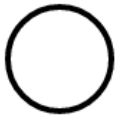
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Text Annotations

- How you document your processes



Start Events

- Start Event 
  - Message Start
  - Timer Start
  - Conditional Start
  - Signal Start

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Some Intermediate Events

Basic Intermediate Event

Message Catch

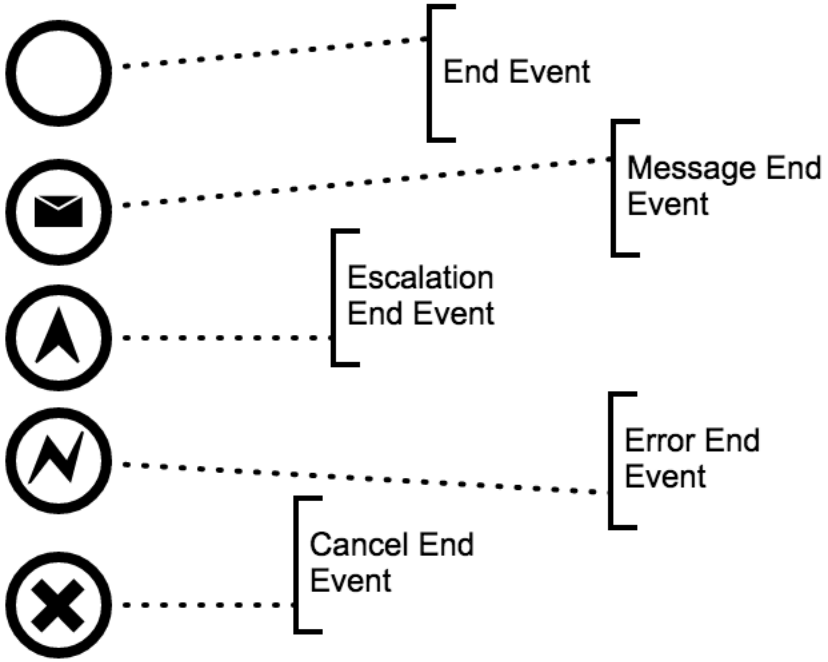
Message Throw

Timer

Escalation

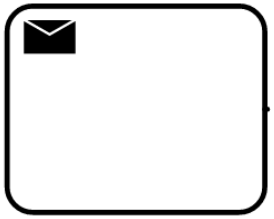
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Some End Events

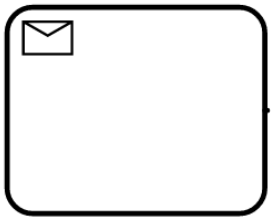


© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

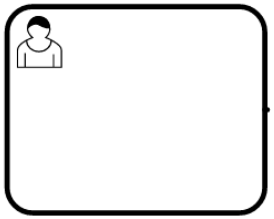
Activities



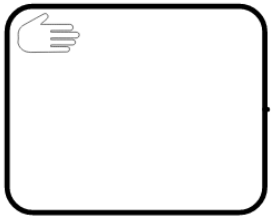
Send Task
- sends a message



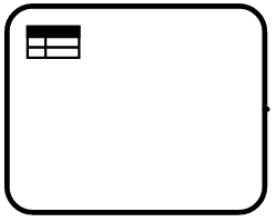
Receive Task
- receives a message



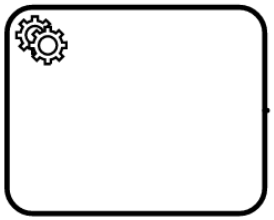
User Task
- requires input or action from a user, mediated by the engine



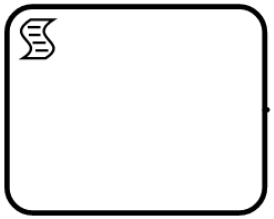
Manual Task
- completely manual task (out side the engine)



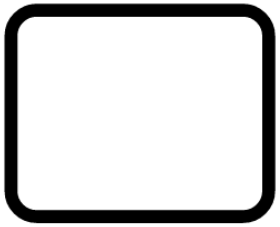
Business Rule Task
- implemented by a BR engine



Service Task
- implemented by a service



Script Task
- implemented by a script



Call Activity-
calls another process

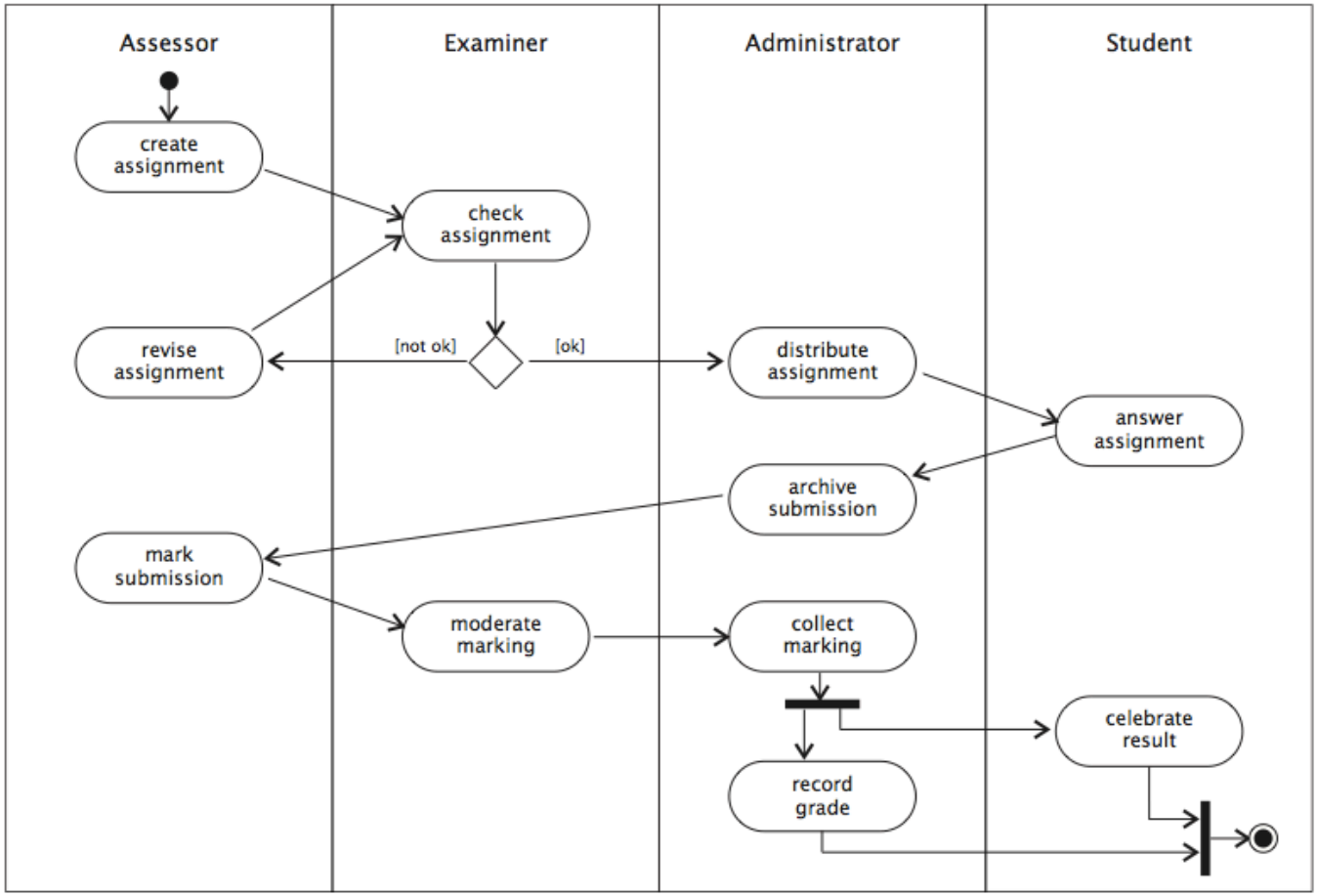
Service Task

- Call a service
  - Unlike BPEL there is no direct way of capturing

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

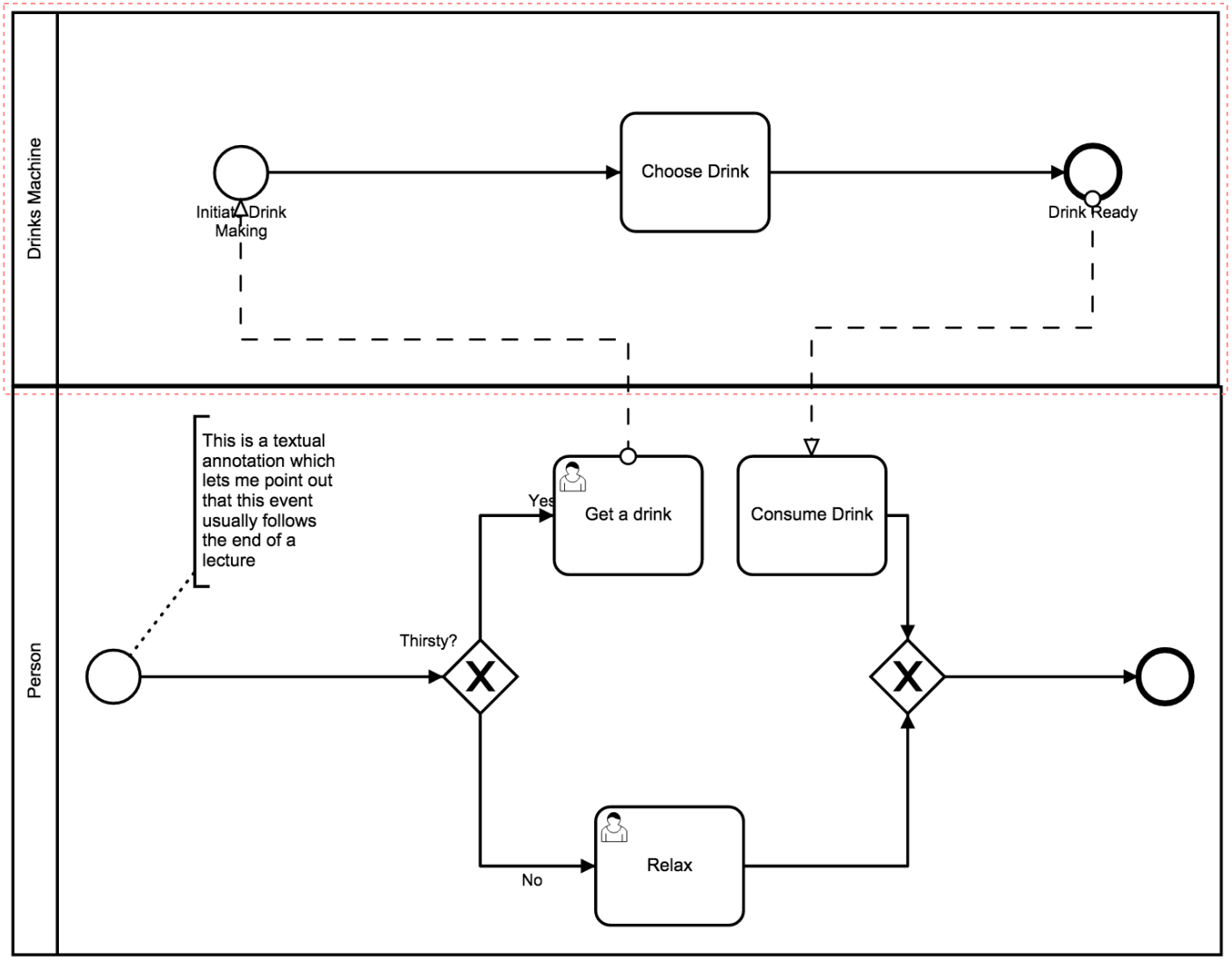
Swimlanes:

partition an activity diagram into the responsibilities of different entities



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Swimlanes represent different participants



or(s).

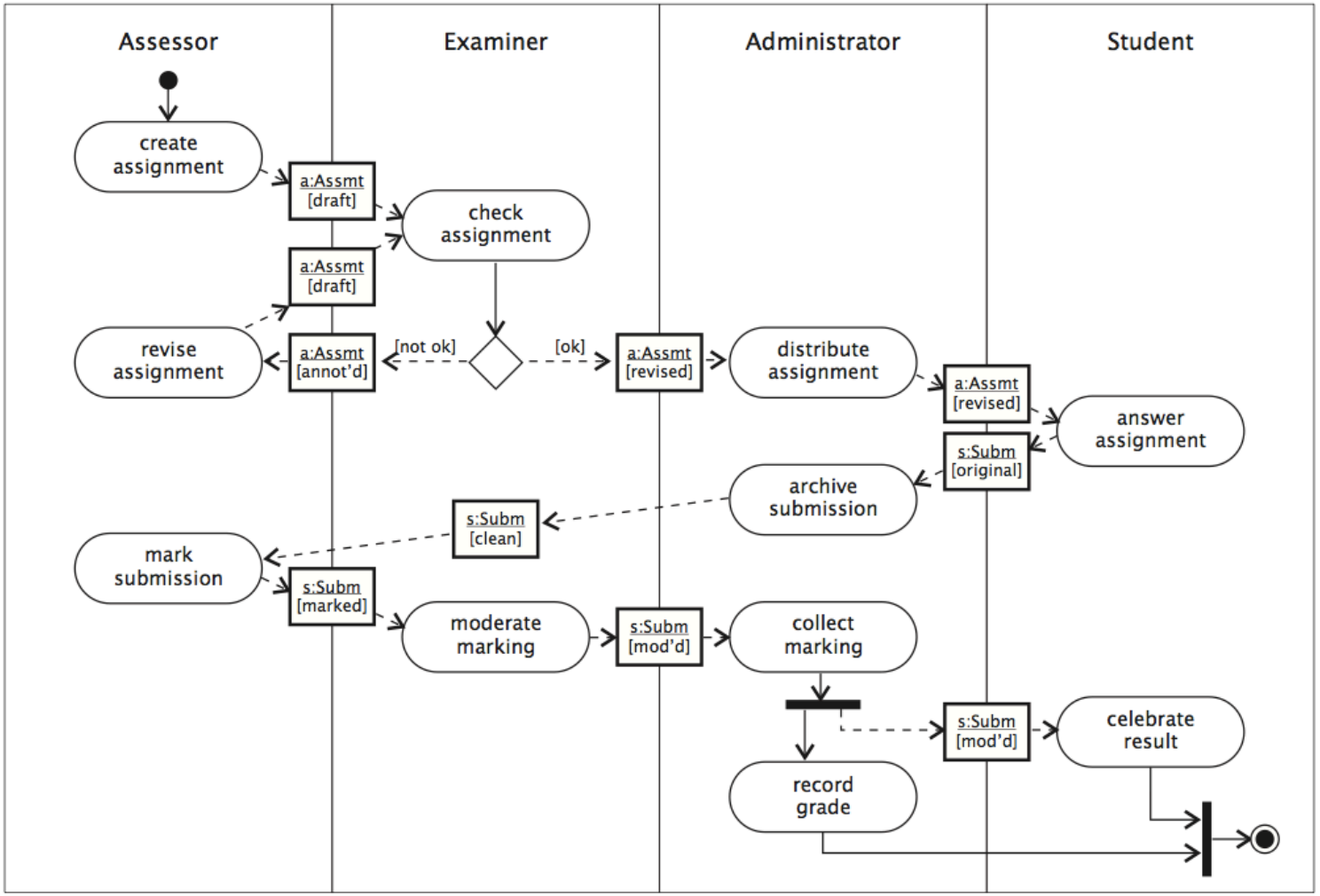
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Data Flow

- Transitions between activities represent *control dependencies*: one activity must complete before another can start
- Workflows also have *data dependencies*: one activity produces a result that another requires
- UML activity diagrams allow *object flow* as well as *control flow*
- Dependent data is shown as an object icon (rectangle with underlined name and type)
  - *dependencies* shown as dashed arrows from generating activity to object, and from object to consuming activity(s)
  - same object may occur multiple times in an activity diagram, typically in different *states* (shown in square brackets after object name)

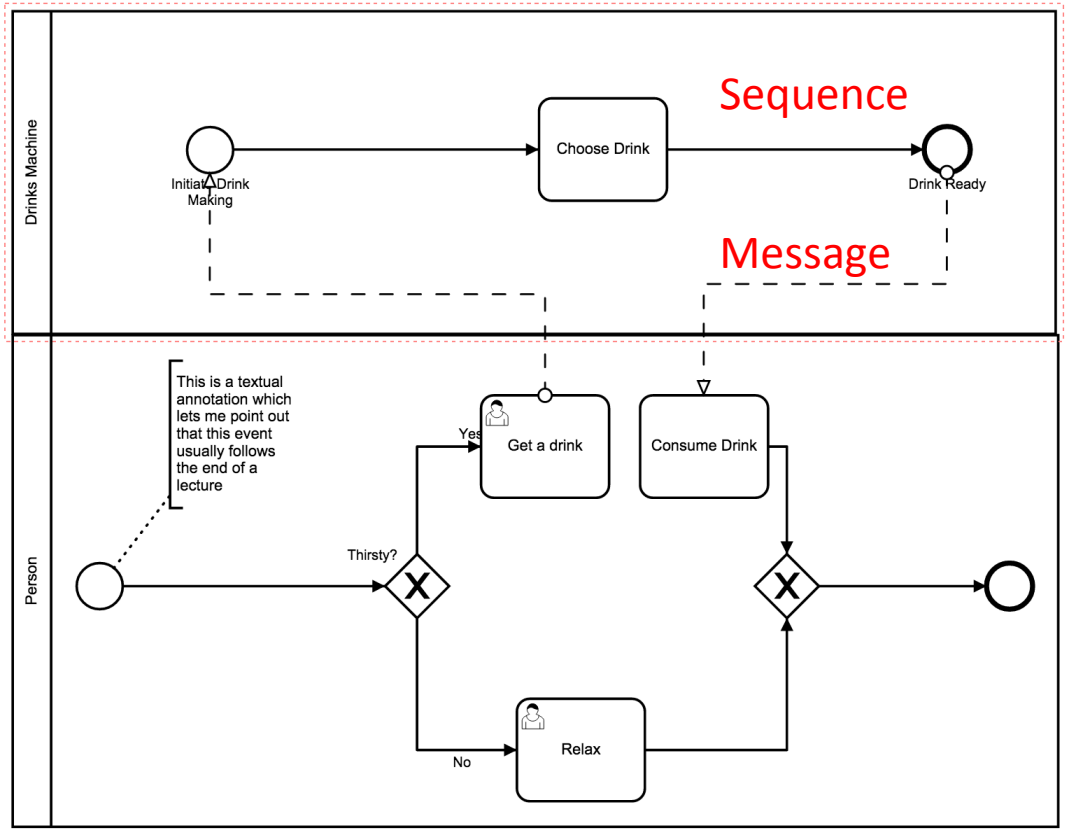
© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Example Object Flow



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Flows



Sequence flows are within a Swimlane

Message flows between swimlanes

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Gateways

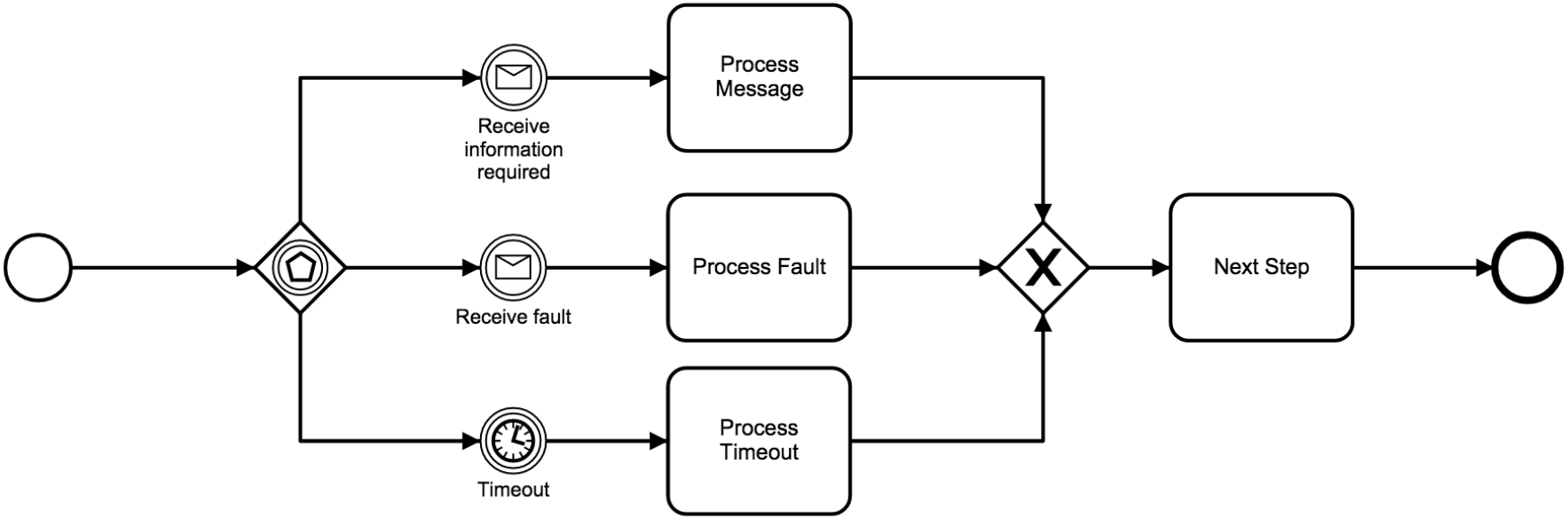
- Exclusive Gateway

  - Fork – choose one path (if/else)
  - Join – wait for a single event
- Parallel Gateway

  - Fork – do both / all paths
  - Join – wait for all inputs

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

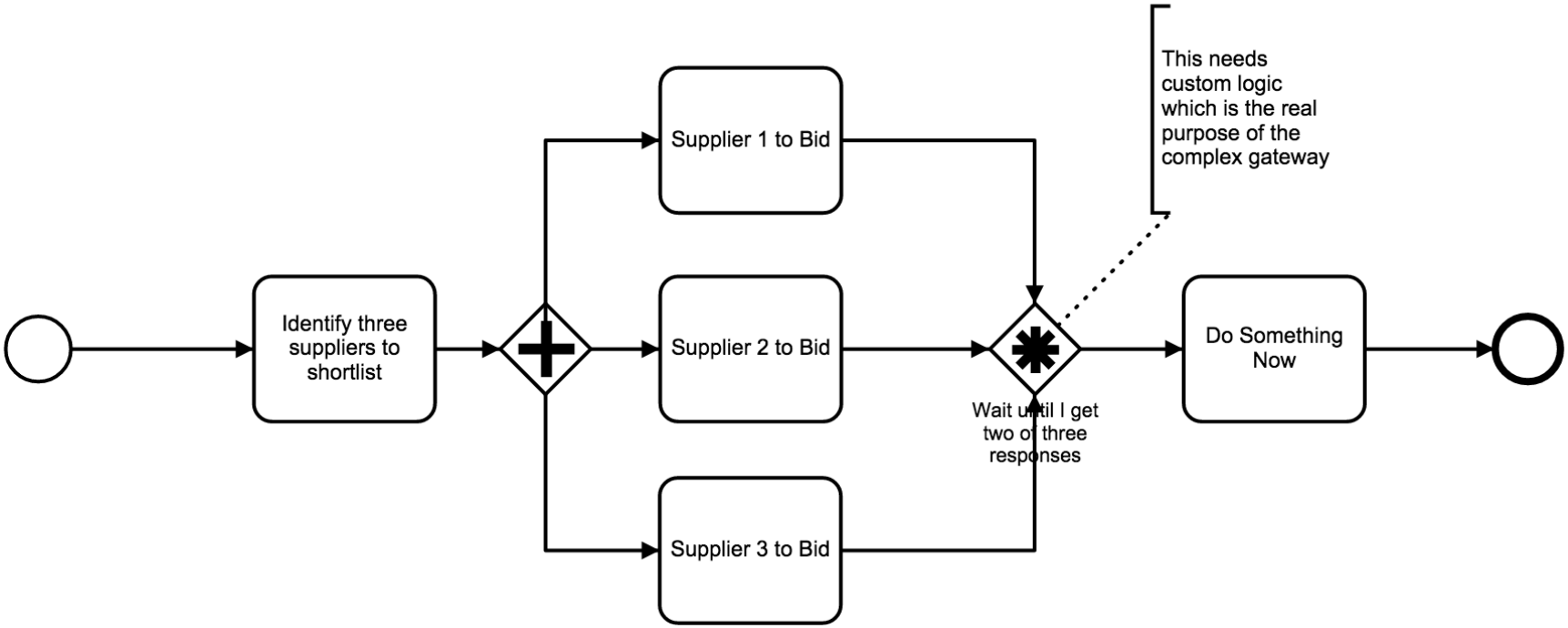
Event Gateway



An Event Gateway allows different events to trigger different actions

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Complex Gateway



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

How much BPMN do you need?

How Much Language is Enough? Theoretical and Practical Use of the Business Process Management Notation
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2038665

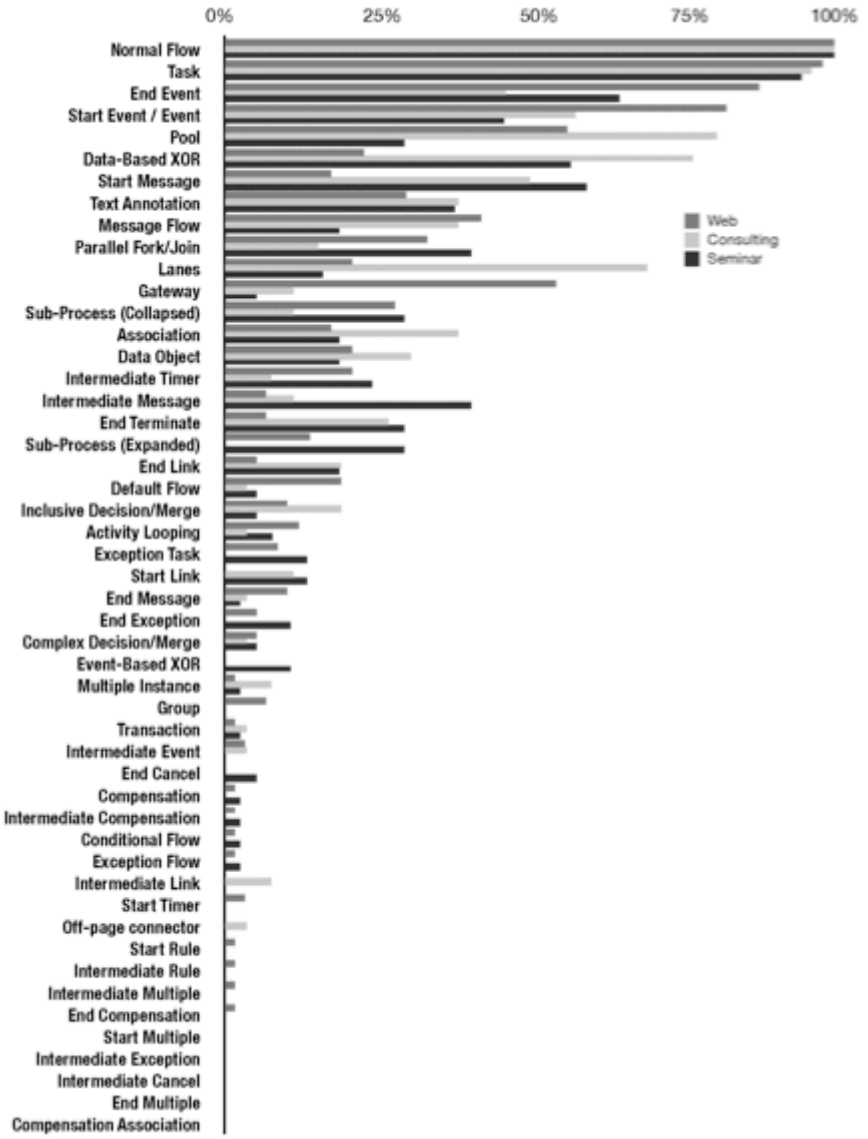
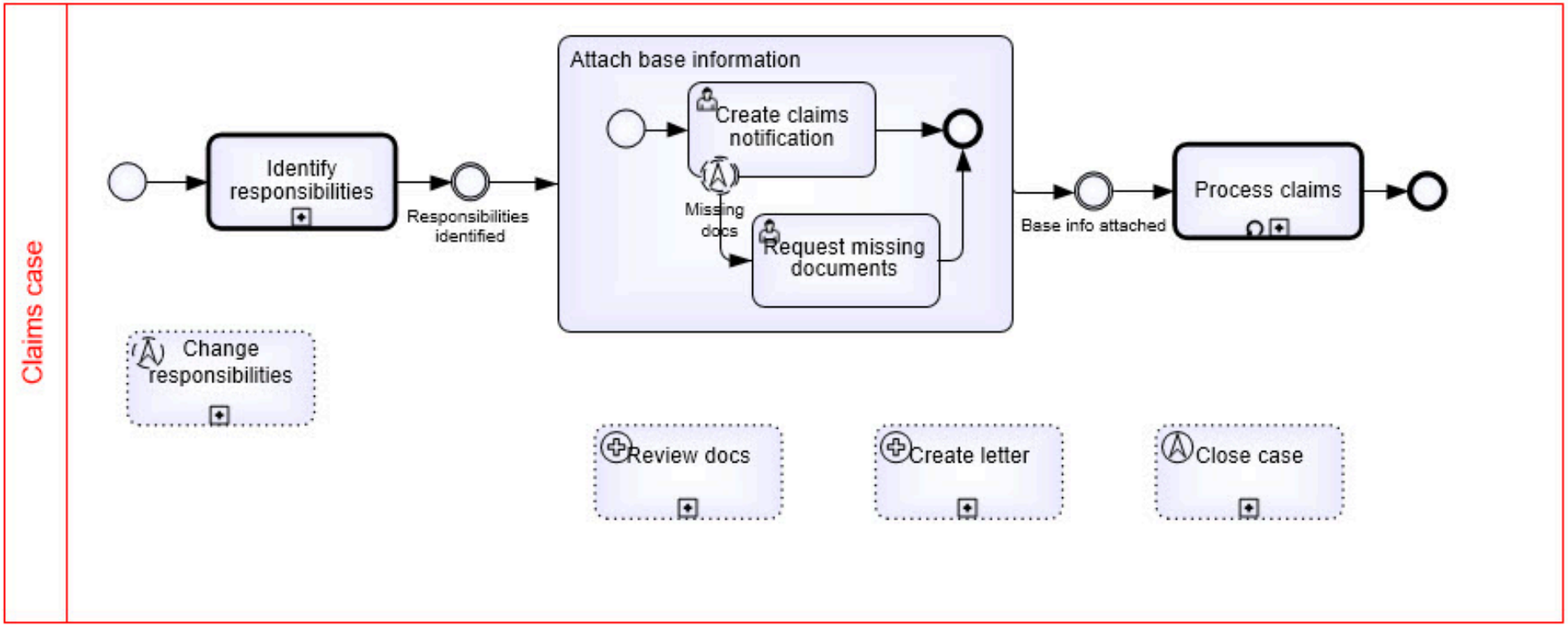


Fig. 1. Occurrence Frequency of BPMN Constructs

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

BPMN Case example

Source: http://brsilver.com/bpmn-cmmn-compared/



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/

Summary

- Process Management has a strong place in composing SOA systems
  - Externalising dependencies
  - Agility
  - Sharing with the business owners
- BPEL is still widely used, but
- BPMN 2.0 is gaining a lot of mindshare
- CMMN also has a smaller but active following
- Other approaches like Amazon SWF may also gain traction

© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See http://creativecommons.org/licenses/by-sa/3.0/