



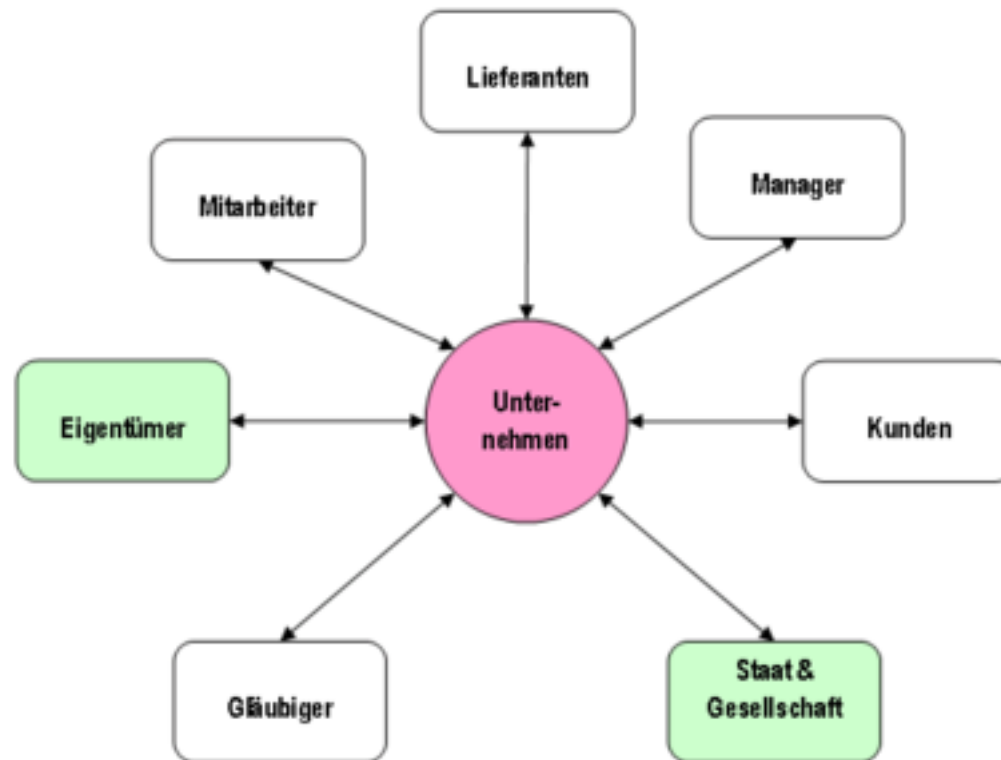
BPMN

- The **Business Process Modeling Notation (BPMN)** is a standardised graphical notation for drawing business processes in a workflow. It is being maintained by the Object Management Group (OMG).
- The primary goal of BPMN is to provide a standard notation that is readily understandable by all business stakeholders.
- BPMN is constrained to support only the concepts of modeling that are applicable to business processes; that i.e. means, organisational structures or data models are not part of the BPMN scope.
- There's four basic element categories to create a Business Process Diagram (BPD):
 - Flow Objects (Events, Activities, Gateways)
 - Connecting Objects (Sequence Flow, Message Flow, Association)
 - Swimlanes (Pool, Lane)
 - Artifacts (Data Objects, Group, Annotation)



Excurs: Stakeholder

- A stakeholder is a person or organization that has a legitimate interest in a project or entity (this only appears in a business context).





BPMN Core Elements

Core Set of BPMN Elements

Flow Objects

Events



Activities



Gateways



Connecting Object

Sequence Flow



Message Flow

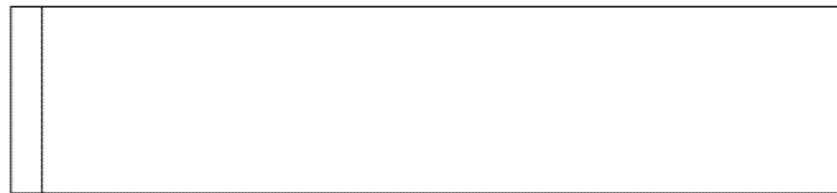


Association

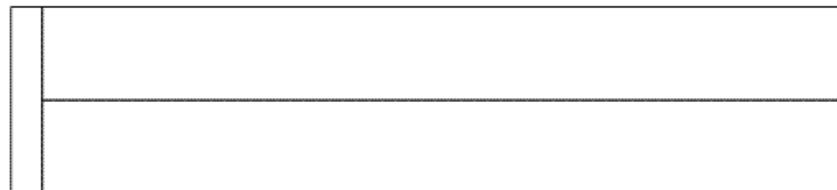


Swimlanes

Pool



Lanes (within a Pool)



Artifacts

Data Object



Name
[State]

Text Annotation

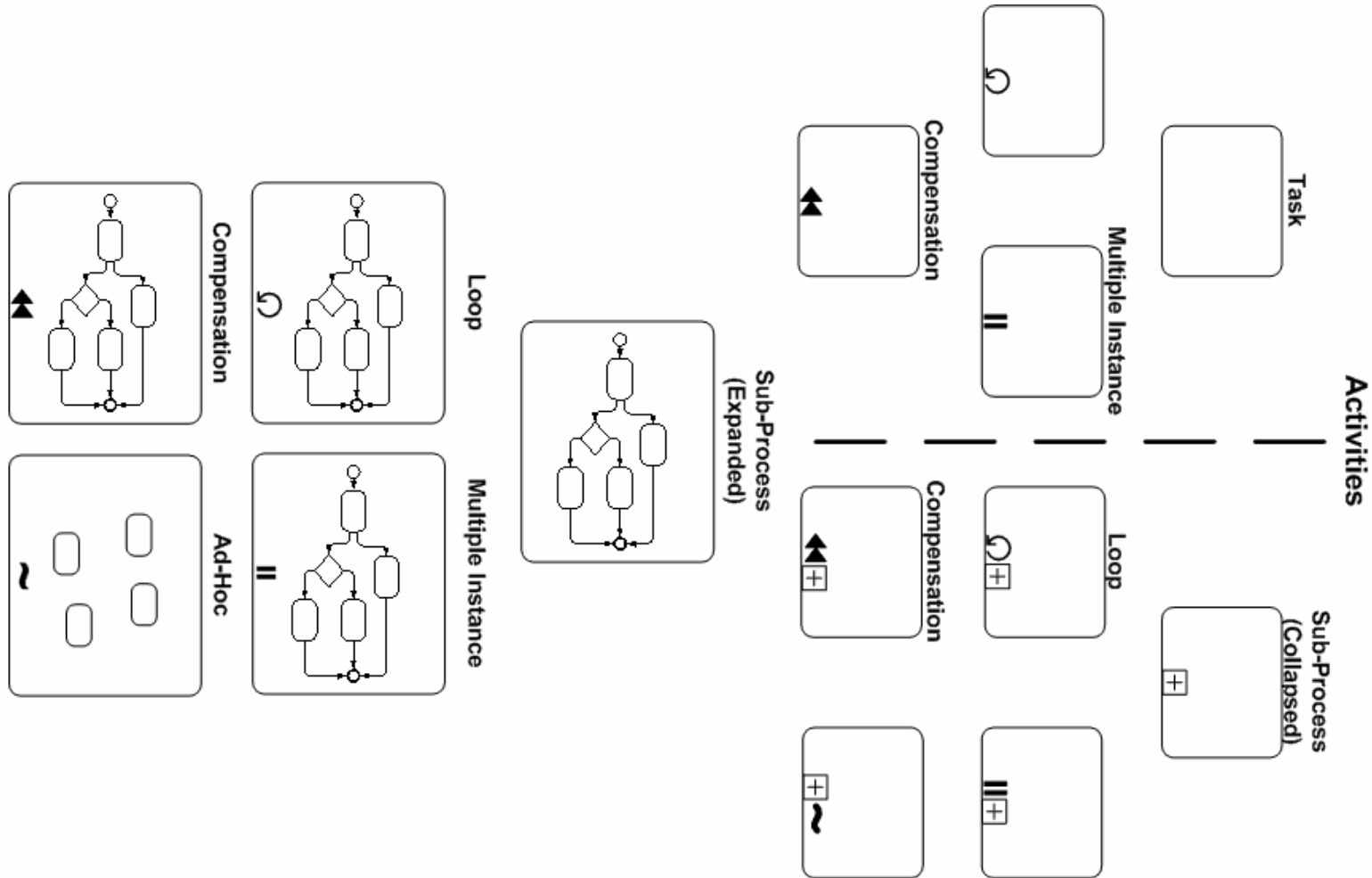
Text Annotation Allows a Modeler to provide additional Information

Group





BPMN Activities





BPMN Events

Events

Start	Intermediate	End

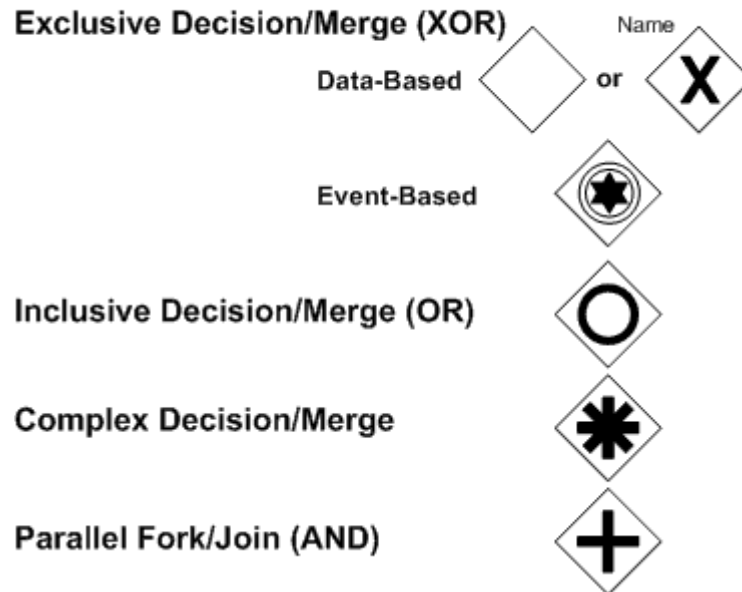
Event Types

Message			
Timer			
Error			
Cancel			
Compensation			
Rule			
Link			
Terminate			
Multiple			



BPMN Gateways

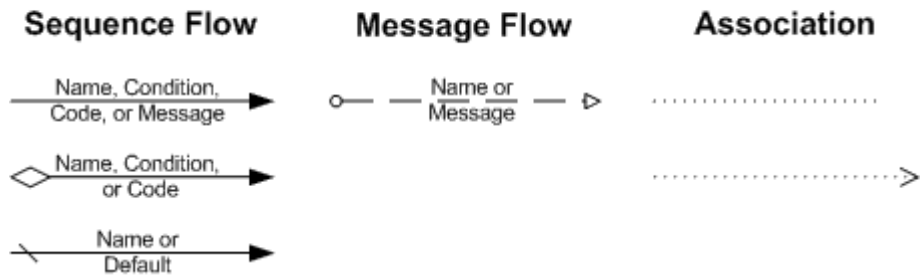
Gateways





BPMN Connections

Connections





BPMN Artifacts

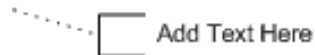
Artifacts

Data Object



Name
[State]

Text Annotation



Add Text Here

Group



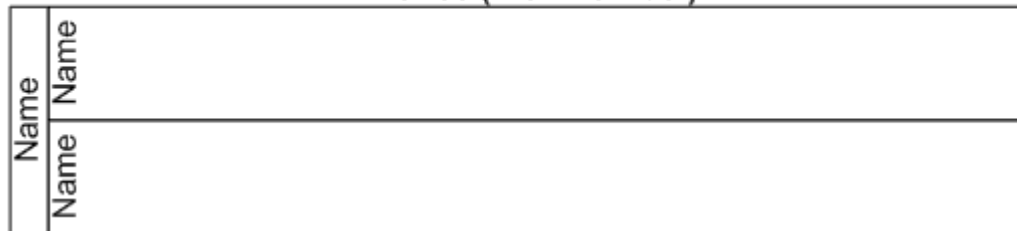


BPMN Swimlanes

Swimlanes Pool

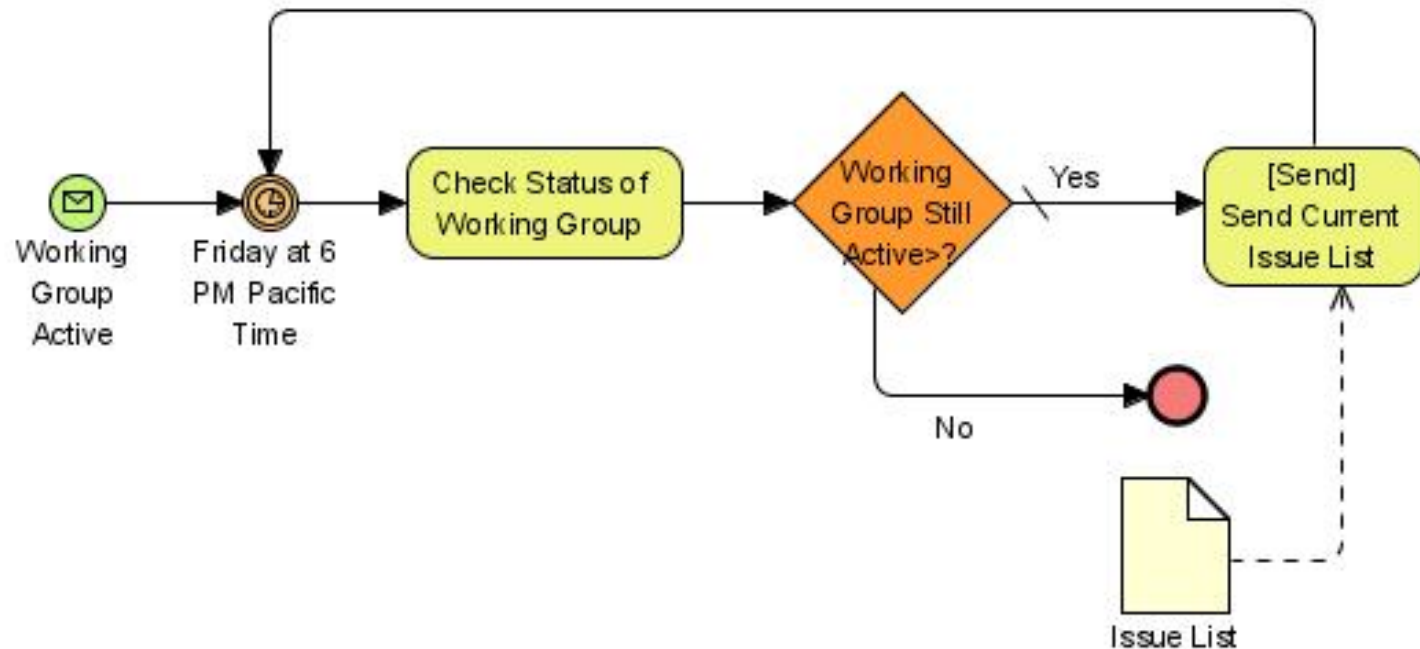


Lanes (within a Pool)



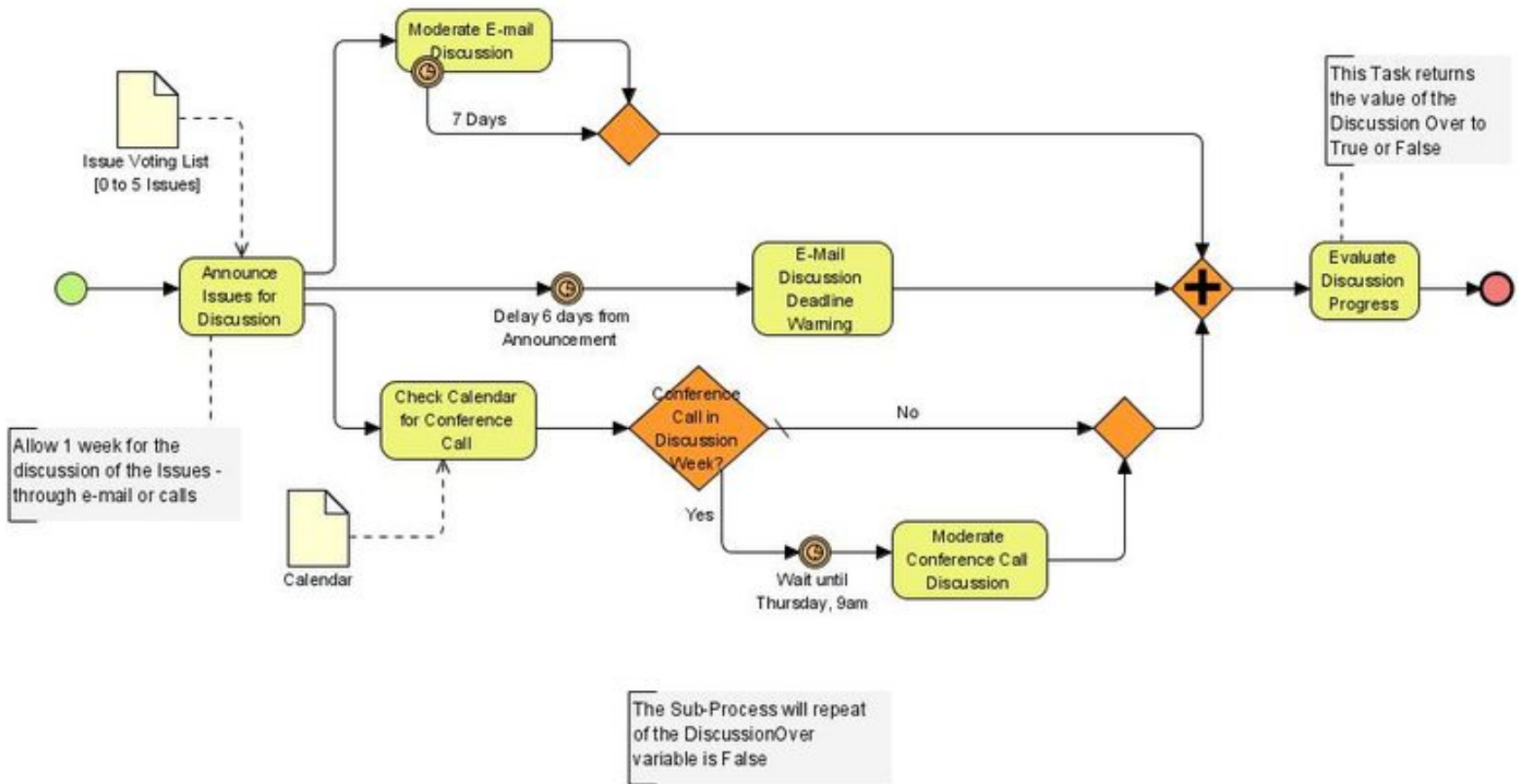


Example I – Normal Flow



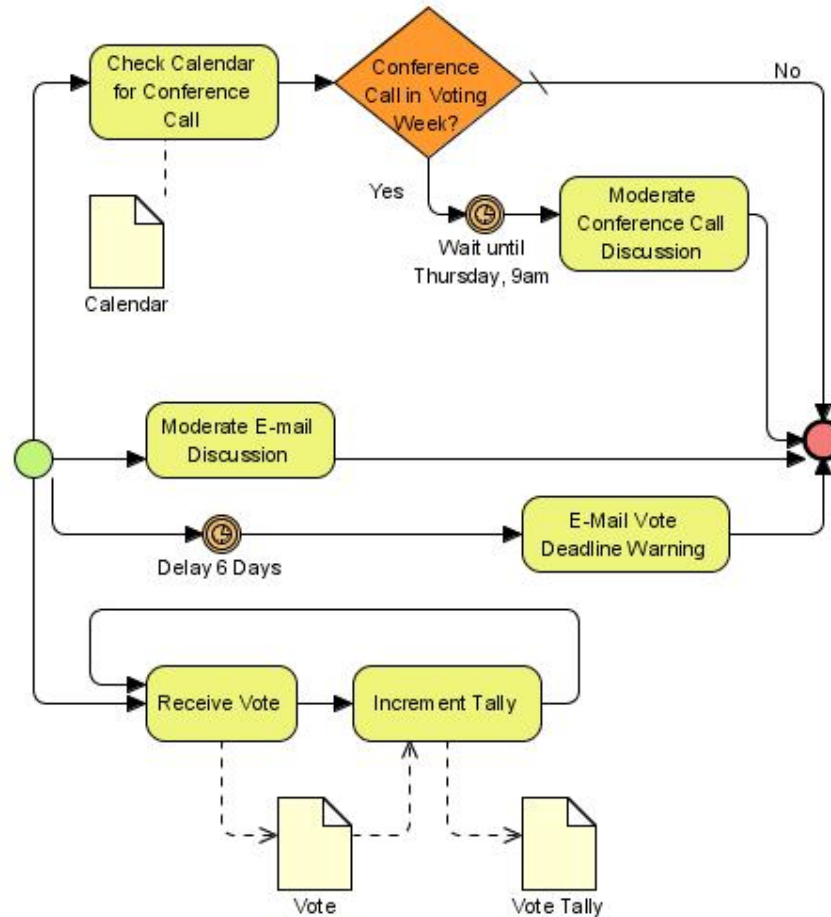


Example II – Discussion Cycle





Example III – Collect Votes





BPMN Process Types

- private processes
- abstract processes
- collaboration processes



Private Processes

- **Private (internal) business processes**

- Private business processes are those internal to a specific organization and are the type of processes that have been generally called workflow or BPM processes. If swim lanes are used then a private business process will be contained within a single Pool. The Sequence Flow of the Process is therefore contained within the Pool and cannot cross the boundaries of the Pool. Message Flow can cross the Pool boundary to show the interactions that exist between separate private business processes.



Abstract Processes

- **Abstract (public) processes**

- These represent the interactions between a private business process and another process or participant. Only those activities that communicate outside the private business process are included in the abstract process. All other “internal” activities of the private business process are not shown in the abstract process. Thus, the abstract process shows to the outside world the sequence of messages that are required to interact with that business process. Abstract processes are contained within a Pool and can be modeled separately or within a larger BPMN diagram to show the Message Flow between the abstract process activities and other entities.



Collaboration Processes

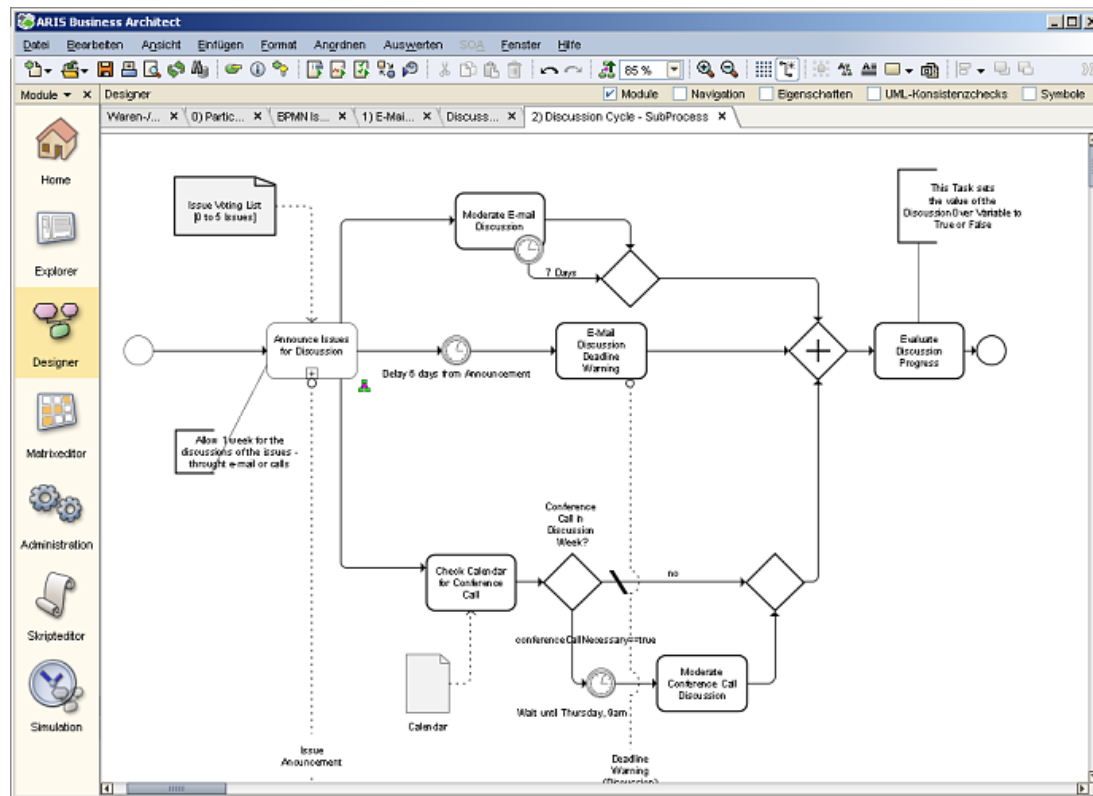
- **Collaboration (global) processes**

- A collaboration process depicts the interactions between two or more business entities. These interactions are defined as a sequence of activities that represent the message exchange patterns between the entities involved. Collaboration processes may be contained within a Pool and the different participant business interactions are shown as Lanes within the Pool. They may also be shown as two or more Abstract Processes interacting through Message Flow (as described in the previous section). These processes can be modeled separately or within a larger BPMN Diagram to show the Associations between the collaboration process activities and other entities.



ARIS BPMN

- offers all elements and content being part of BPMN
- converts BPMN \leftrightarrow EPK / backwards (if allowed) and checks semantics





Resources

- Object Management Group – Business Process Management Initiative
 - <http://bpmn.org/>
- BPMN Supporters
 - http://bpmn.org/BPMN_Supporters.htm
- BPMN 1.0 OMG Specification
 - <http://bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>
- BPMN Introduction, Survey & Training Kit
 - <http://www.diveintobpm.org/>



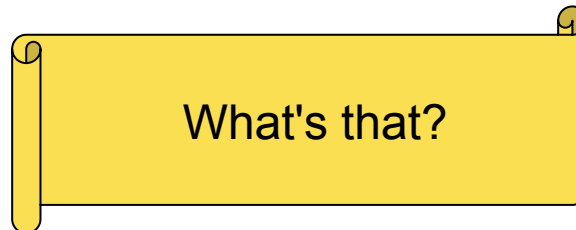
Tools

- Intalio Business Process Management Suite (Eclipse-based)
 - <http://bpms.intalio.com/index.php>
- Soyatec eBPMN (Eclipse-based)
 - <http://www.soyatec.com/ebpmn/features.html>
- Eclipse STP BPMN Modeler
 - <http://www.eclipse.org/stp/bpmn/>



BPEL

- The **Business Process Execution Language (BPEL)** is an executive business process modeling language. It is serialised via XML and describes processes whose activities are implemented through web services.
- BPEL cannot interact directly with human resources, but only with web services, which then can take the part of interfaces to humans.
- The initiators are Microsoft, IBM and BEA.
- BPEL itself doesn't describe a way for graphical illustrations of business processes; this is part of BPMN.
- BPEL was designed for "Programming in the Large".





Executable vs. Abstract Processes

- executable processes
 - can be deployed on a workflow engine and then be run
- abstract processes
 - are views on executable processes, to describe the latter ones (so-called behavioral interfaces)
 - are used to hide the details of a process from e.g. business partners



BPEL4People / Subprocesses

- BPEL4People
 - IBM and SAP created an initiative called BPEL4People to collect specifications for interaction with humans. So far only a whitepaper exists.
- WS-BPEL 2.0 Extension for Subprocesses
 - BPEL doesn't know subprocesses. This results in difficulties, as BPEL's not intended to terminate a subprocess when the main one gets terminated. Such functionality has to be implemented by a further specific web service (which then is not part of the BPEL standard).
 - Once again IBM and SAP introduced a whitepaper named "WS-BPEL 2.0 Extension for Subprocesses" that explains an extension for subprocesses within BPEL.



BPEL – Elements (I)

- `<receive>` / `<reply>`
 - Warten auf Nachricht (`<receive>`) und Antwort auf eine Nachricht (`<reply>`). Die Kombination beider Aktivitäten ermöglicht eine request - response Operation für einen WSDL portType des Prozesses.
- `<invoke>`
 - Aufruf eines Web Services.
- `<assign>`
 - Updaten von Variablen mit neuen Daten.
- `<throw>` / `<catch>`
 - Interne Fehler eines Geschäftsprozesses signalisieren (`throw`) bzw. abfangen (`catch`).
- `<terminate>`
 - Explizites Beenden des Prozesses.



BPEL – Elements (II)

- `<wait>`
 - Der Prozess wartet eine definierte Zeit.
- `<empty>`
 - Leere Aktivität - wird verwendet, um Fehler abzufangen oder um Aktivitäten eines Geschäftsprozesses zu synchronisieren.
- `<sequence>`
 - Sequentielle Ausführung der nachgereihten Aktivitäten.
- `<switch>`
 - Aktivität zur Modellierung von Verzweigungen. Kann eine oder mehrere Bedingungen (`<case condition="boolscher Ausdruck">`) und optional einen alternativen Zweig (`<otherwise>`) enthalten.



BPEL – Elements (III)

- `<while>`
 - Mittels `<while>` wird die betroffene Aktivität so lange wiederholt ausgeführt, bis der boolesche Ausdruck nicht mehr erfüllt wird.
- `<pick>`
 - Diese Aktivität wartet auf das Auftreten eines Ereignisses aus einer Menge von Ereignissen und führt danach die entsprechend angeführte Aktivität aus.
- `<flow>`
 - Das `<flow>` Konstrukt ermöglicht die parallele Verarbeitung von einer oder mehreren Aktivitäten. Mittels des `<link>` - Konstruktes können direkt oder indirekt eingebettete Aktivitäten synchronisiert werden.



BPEL – Elements (IV)

- `<scope>`
 - Das `<scope>` - Konstrukt ermöglicht die Verbindung einer eingebetteten Aktion mit Variablen, einer Fehler- und Kompensationsbehandlung. Zur Koordination wird hierbei das BusinessAgreement Protokoll der WS-Transaction Spezifikation verwendet.
- `<compensate>`
 - `<compensate>` wird verwendet, wenn eine in einen `<scope>` eingebettete Aktion erfolgreich durchgeführt wurde, diese Aktion aber kompensiert werden muss. Dieses Konstrukt kann nur von einem anderen compensation handler oder vom fault handling aufgerufen werden.



Resources

- WS BPEL 2.0 Specification
 - <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- Advanced BPEL – Oracle SOA Best Practices
 - http://www.oracle.com/technology/pub/articles/bpel_cookbook/index.html
- Pattern-based Evaluation of Oracle BPEL
 - http://is.tm.tue.nl/research/patterns/download/Oracle_BPEL_v.10.1.2.pdf
- Article: BPEL & Java
 - <http://www.theserverside.com/tt/articles/article.tss?l=BPELJava>
- Dumas, M.; van der Aalst, W.M.P., ter Hofstede, A.H.M. (editors):
Process-Aware Information Systems: Bridging People and Software
Through Process Technology. Wiley and Sons, 2005.



Tools

- Active BPEL Open Source Engine
 - <http://www.active-endpoints.com/active-bpel-engine-overview.htm>
- NetBeans Enterprise Pack with BPEL Support
 - <http://www.netbeans.org/products/enterprise/>
- Apache ODE (Orchestration Director Engine)
 - <http://ode.apache.org/>
- Queensland University of Technology – Translators (BPMN to BPEL, BPMN to Petri Nets, BPEL to Petri Nets)
 - <http://www.bpm.fit.qut.edu.au/projects/babel/tools/>