

Process modeling

PV207 – Business Process Management

Spring 2015

Jiří Kolář

Last lecture recap

- Why we need specific methodologies and methods for BPM-based development?

Last lecture recap

- Why we need specific methodologies and methods for BPM-based development?
- Top-down vs bottom-up approach

Last lecture recap

- Why we need specific methodologies and methods for BPM-based development?
- Top-down vs bottom-up approach
- Business analysis
 - Vision and mission

Last lecture recap

- Why we need specific methodologies and methods for BPM-based development?
- Top-down vs bottom-up approach
- Business analysis
 - Vision and mission
 - Goals and objectives

Last lecture recap

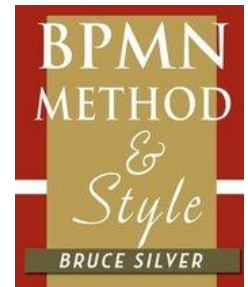
- Why we need specific methodologies and methods for BPM-based development?
- Top-down vs bottom-up approach
- Business analysis
 - Vision and mission
 - Goals and objectives
 - Metrics, KPIs, KRIs

Lecture overview

- Why modeling?
- Process development roles
- Modeling notations
- Workflow modeling
- BPMN 1.1
- BPEL
- BPMN 2.0
- BPMN 2.0
 - Object classes
 - Activities
 - Events
 - Gateways
 - Connecting objects
 - Artifacts
 - Process types
 - Examples

BPMN 2.0 Information sources

- BPMN method and style Bruce Silver,
 - ISBN 20099780982368107
 - 30\$, kindle 13,80\$
- BPMN 2.0 poster
 - http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf
- Signavio modeler – academic licence
 - <http://academic.signavio.com/p/login>
- BPMN official OMG website
 - <http://www.bpmn.org/>



Activities

- Task**: A Task is a unit of work, the job to be performed. When marked with a symbol it indicates a Sub-Process, an activity that can be refined.
- Transaction**: A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.
- Event Sub-Process**: An Event Sub-Process is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.
- Call Activity**: A Call Activity is a wrapper for a globally defined Task or Process reused in the current Process. A call to a Process is marked with a symbol.

Activity Markers

Markers indicate execution behavior of activities:

- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker

Task Types

Types specify the nature of the action to be performed:

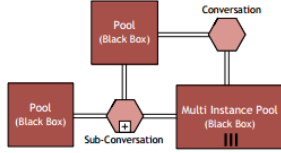
- Send Task
- Receive Task
- User Task
- Manual Task
- Business Rule Task
- Service Task
- Script Task

- Sequence Flow**: defines the execution order of activities.
- Default Flow**: is the default branch to be chosen if all other conditions evaluate to false.
- Conditional Flow**: has a condition assigned that defines whether or not the flow is used.

Conversations

- A Conversation defines a set of logically related message exchanges. When marked with a symbol it indicates a Sub-Conversation, a compound conversation element.
- A Call Conversation is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Sub-conversation is marked with a symbol.
- A Conversation Link connects Conversations and Participants.

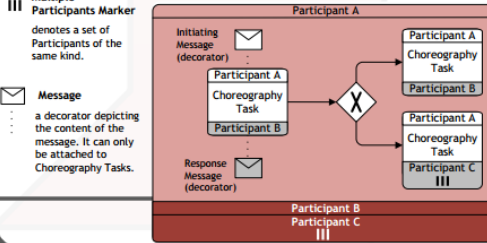
Conversation Diagram



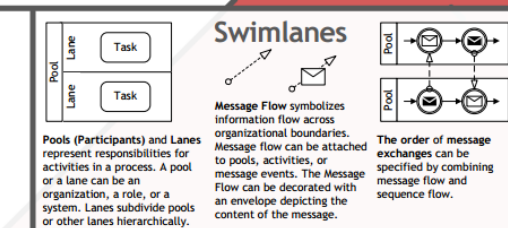
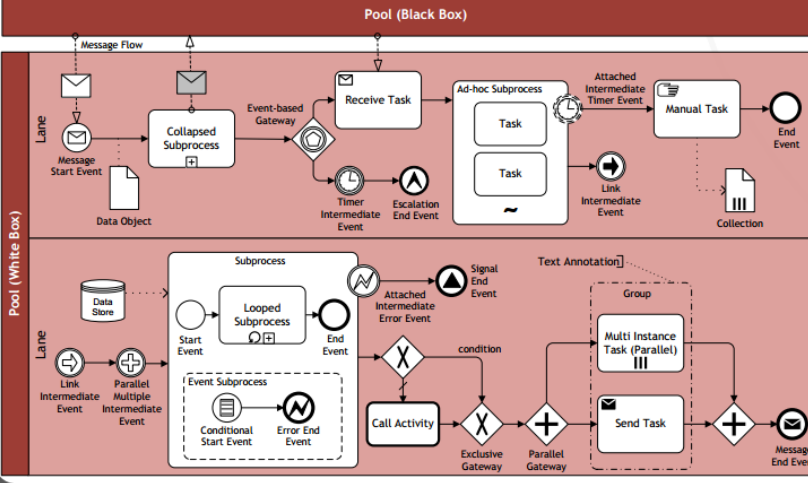
Choreographies

- Participant A**: Choreography Task
 - Participant B**: Sub-Choreography
 - Participant A**: Call Choreography
 - Participant B**: Sub-Choreography
 - Participant C**: Sub-Choreography
- A Choreography Task represents an Interaction (Message Exchange) between two Participants.
- A Sub-Choreography contains a refined choreography with several Interactions.
- A Call Choreography is a wrapper for a globally defined Choreography Task or Sub-Choreography. A call to a Sub-Choreography is marked with a symbol.

Choreography Diagram



Collaboration Diagram



Events

| | Start | Intermediate | End |
|------------------------------------|-------|--------------|-----|
| Standard | | | |
| Event Sub-Process Interrupting | | | |
| Event Sub-Process Non-Interrupting | | | |
| Catching | | | |
| Boundary Interrupting | | | |
| Boundary Non-Interrupting | | | |
| Throwing | | | |
| Standard | | | |

None: Untyped events, indicate start point, state changes or final states.

Message: Receiving and sending messages.

Timer: Cyclic time events, points in time, time spans or timeouts.

Escalation: Escalating to an higher level of responsibility.

Conditional: Reacting to changed business conditions or integrating business rules.

Link: Off-page connectors. Two corresponding link events equal a sequence flow.

Error: Catching or throwing named errors.

Cancel: Reacting to cancelled transactions or triggering cancellation.

Compensation: Handling or triggering compensation.

Signal: Signalling across different processes. A signal thrown can be caught multiple times.

Multiple: Catching one out of a set of events. Throwing all events defined.

Parallel Multiple: Catching all out of a set of parallel events.

Terminate: Triggering the immediate termination of a process.

Data

- A Data Object represents information flowing through the process, such as business documents, e-mails, or letters.
- A Collection Data Object represents a collection of information, e.g., a list of order items.
- A Data Input is an external input for the entire process. A kind of input parameter.
- A Data Output is data result of the entire process. A kind of output parameter.
- A Data Association is used to associate data elements to Activities, Processes and Global Tasks.
- A Data Store is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.

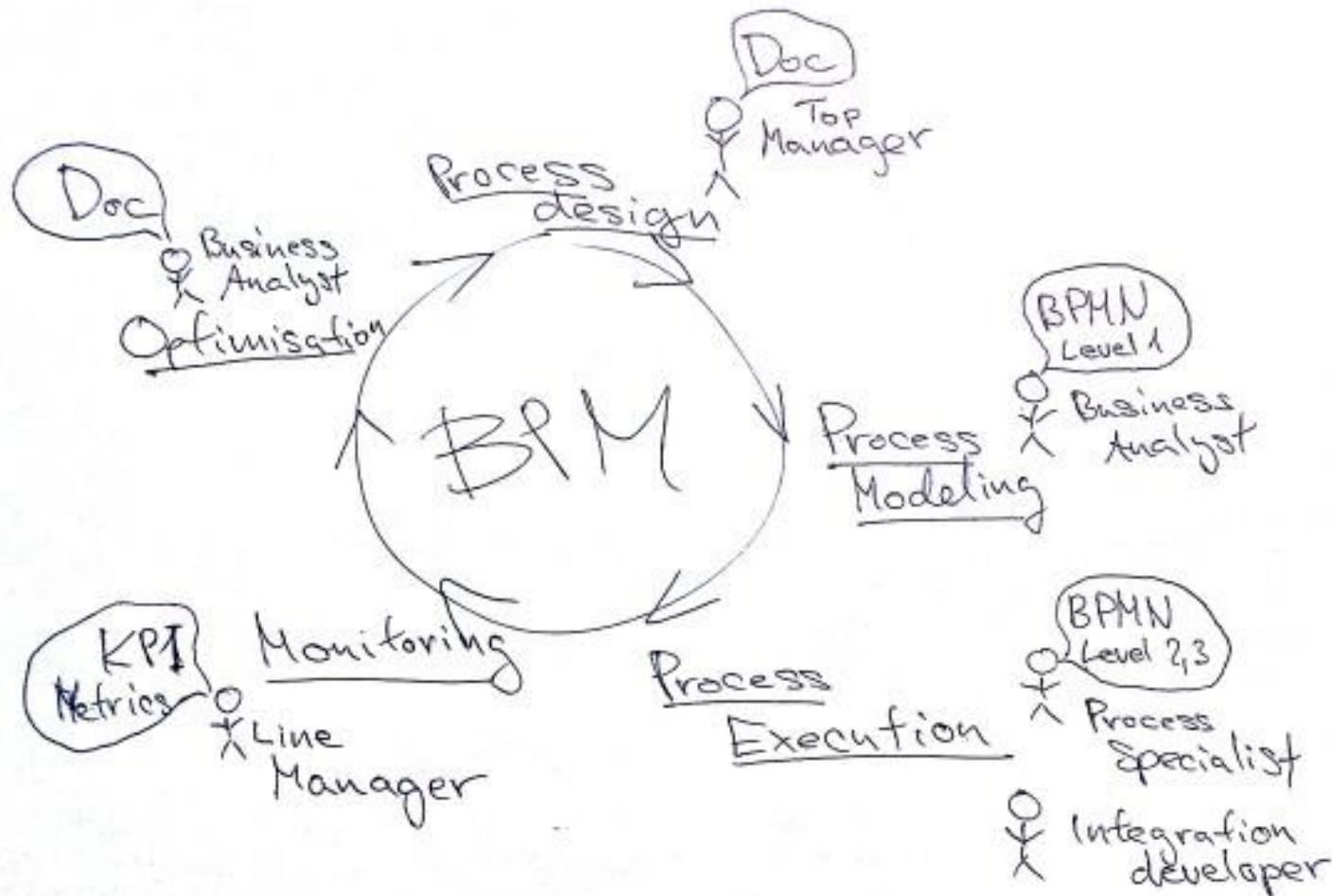
Why process modeling?

- Elegant way to express structure of a process
- Visual models are **easily understandable by all participants** of the development cycle
- Minimise the misunderstandings during the transformation **from analytical description** to the **executable implementation** of the process
- Covers **nested structure** (sub-processes)
- Covers inter-process/inter-system **interactions**
- Pictures are fun

Roles in development cycle

- **Business Analyst**
 - Sum business strategy
 - Describe goals & objectives, KPIs
 - Describe processes
 - Design BPMN diagrams (Level 1)
- **Process specialist**
 - Design BPMN diagrams (Level 1,2,3)
 - Design monitoring models
- **Process developer**
 - Detail BPMN Level 3
 - Implement services and deploy processes

Roles in development cycle



Modeling notations

- **BPMN 1.0 – 1.1**
 - Analytical modeling, not tight with semantics, not executable
 - XPDL semantics
- **BPEL**
 - Technical modeling, very detailed
 - service orchestration, executable
- **BPMN 2.0**
 - Analytical modeling (Level 1,2)
 - Defined semantic – executable (Level 3)

BPMN 2.0 Three levels

- Level 1 (Structure)
 - Captures basic structure of the process
 - Business experts \Leftrightarrow analysts/developers
- Level 2 (Analytical)
 - More details of process behaviour (interactions, events, timing)
 - Process analysts \Leftrightarrow Process developers
- Level 3 (Executable)
 - Specifies all used services and activity tasks
 - “”(Process developers \Leftrightarrow Process engine) “”

BPMN 2.0 Three levels

- Level 1 (Structure)
 - Captures basic structure of the process
 - Business experts \Leftrightarrow analysts/developers
- Level 2 (Analytical)
 - More details of process behaviour (interactions, events, timing)
 - Process analysts \Leftrightarrow Process developers
- Level 3 (Executable)
 - Specifies all used services and activity tasks
 - “”(Process developers \Leftrightarrow Process engine) “”

Model quality aspects

- **Validity against BPMN specification**
 - Wrong connections of the flow
 - Missing start/end
 - Wrong used gateways
- **Model understandability**
 - Reasonable naming of activities
 - Reasonable amount of connections/gateways/activities
- **Expressiveness**
 - How it reflects the situation in real world
 - Granularity of activities
- **Compliance to the modeling best practices**
 - Modeling style (Seminars & third modeling lecture)

Homework Assignments

- This week seminar:
 - L1 Homework assignment **deadline this friday**
- Next week seminar:
 - You receive corrected homeworks
 - L2 Homework assignment
- Homework submission
 - Submit **printed version** to the box called "PV207" next to entrance to room D1, **before deadline**
 - Submit **electronic version**
 - Export to PNG in Signavio, submit to IS MUNI to folder "homeworks" as <surename>_bpmn.png , **before deadline**

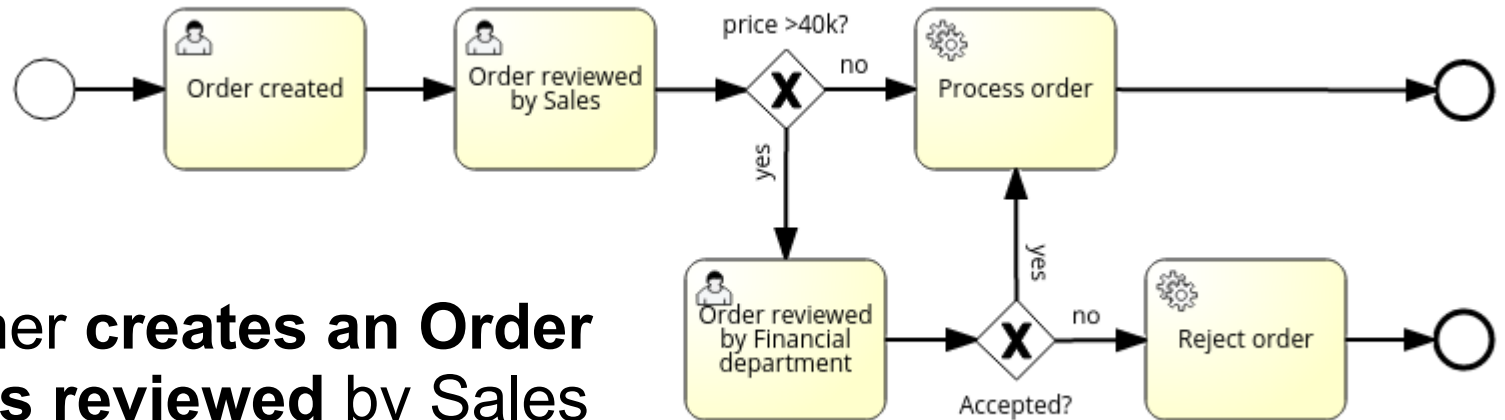
Questions?
Break 10mins
Feedback

BPMN 2.0 (Level 1)

Object classes

- Flow Objects
 - Event
 - Activity
 - Gateway
- Connecting Objects
 - Sequence Flow
 - Message Flow
 - Association
- Artifacts
 - Data Object
 - Group
 - Annotation
- Swimlanes
 - Pool
 - Lane

BPMN 2.0 – Process example



1. Customer **creates** an **Order**
2. **Order is reviewed** by Sales

2.1. If price of the Order is **lower** than 40 000\$, it is accepted

2.2. If price is **over** 40 000\$ it have to be confirmed by Financial department

2.3. Order can be rejected by the department

3. Otherwise the order is processed

Flow object - Activity

- Represent certain step, an activity
- Types of activity

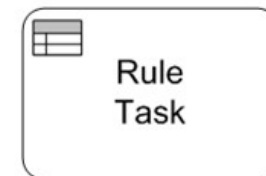
- Atomic activity = Task






- Complex activity = Subprocess



- Types of task



Connecting Objects

- Process sequence flow 
 - Define order of activities
- Message flow 
 - **Does not influence the process flow !**
 - Message flow between two objects
- Association 
 - **Does not influence the process flow !**
 - Connect objects with artifacts (labels,data objects..)

Flow object - Event

- Represent event that occur in a process
- Have impact on process flow

- Types

- Start



- Intermediate



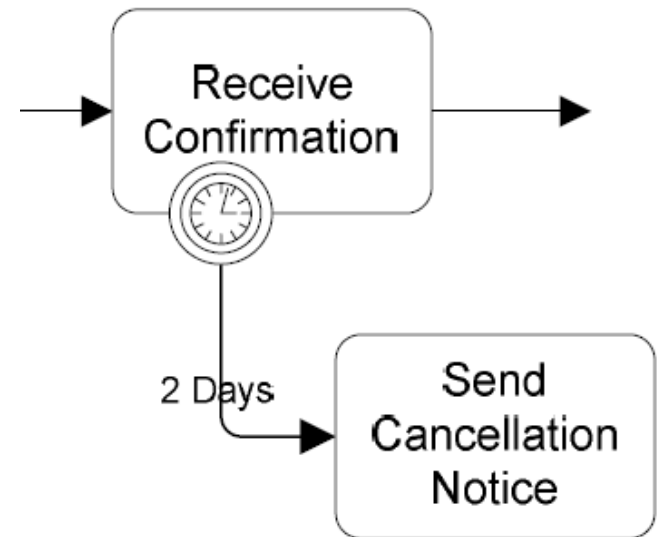
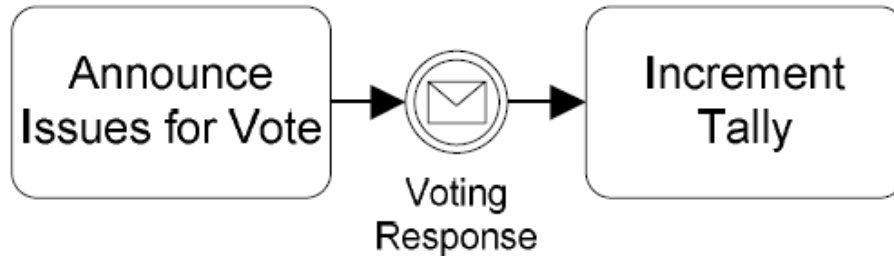
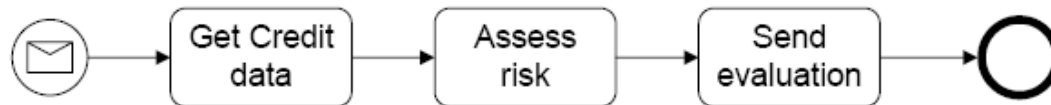
- End









- Extended (Level 2)



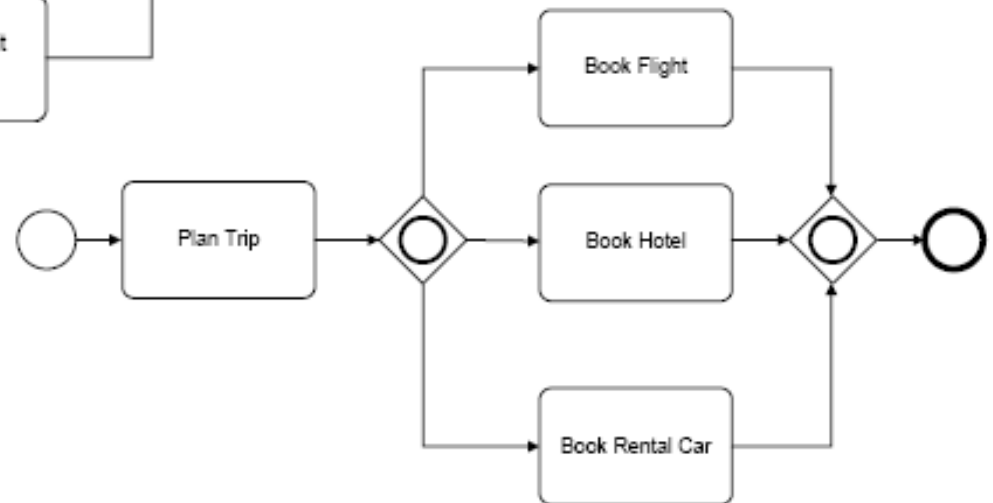
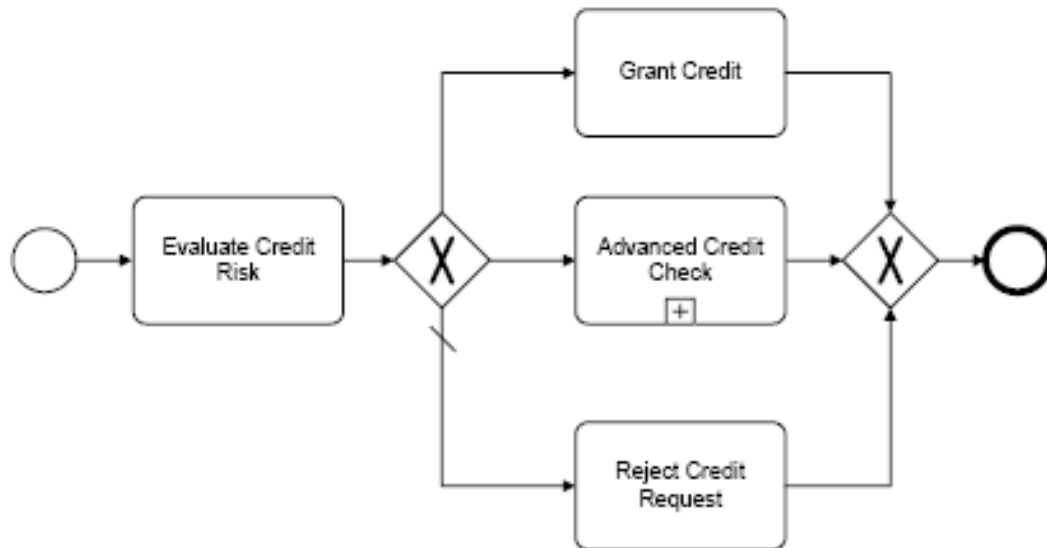
Flow objects - examples



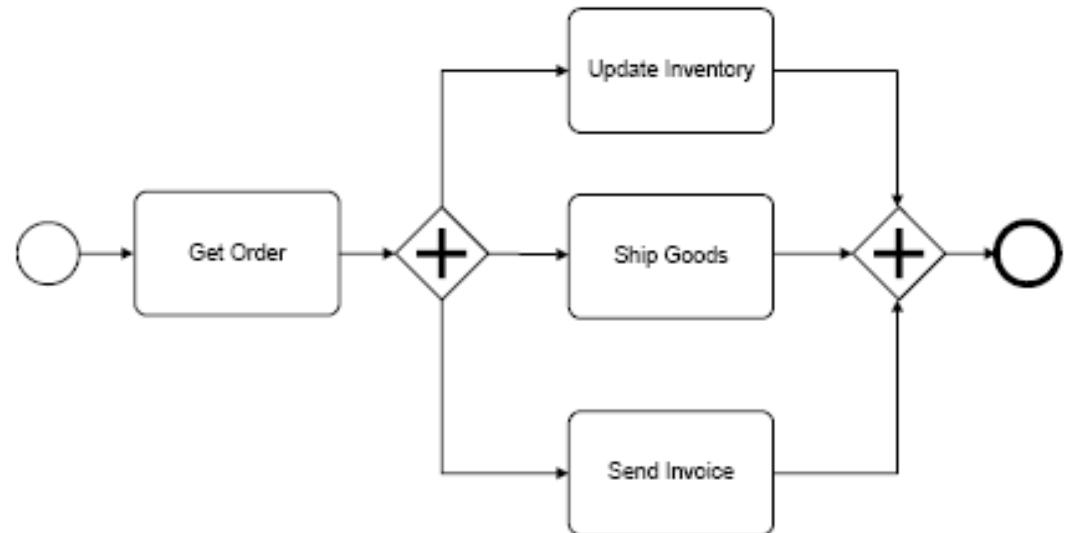
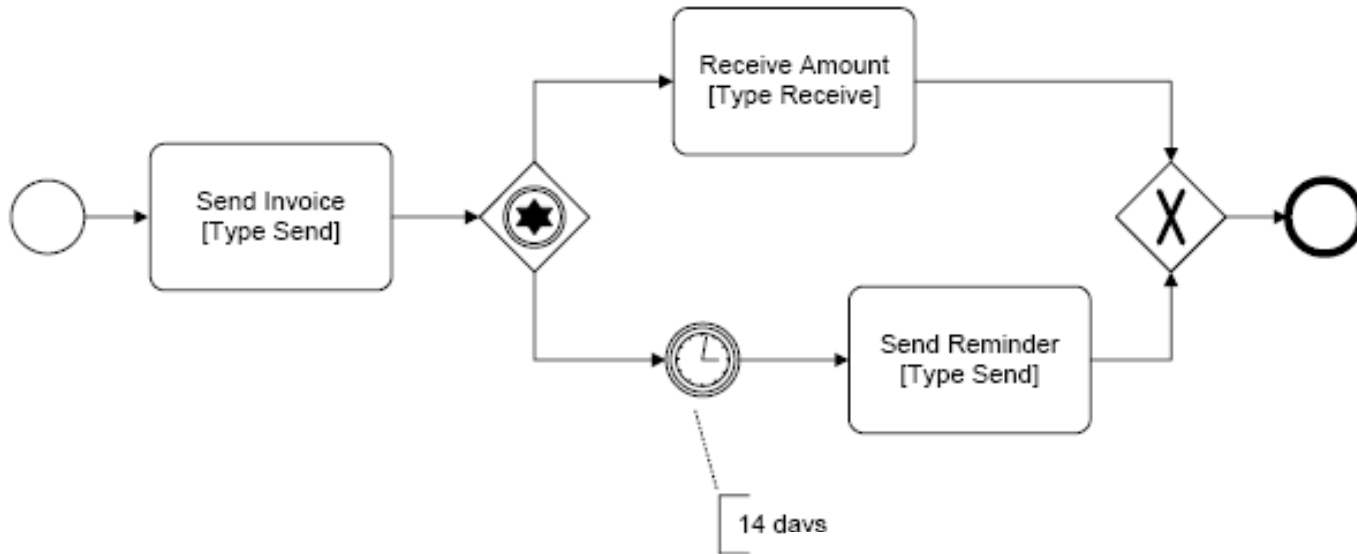
Flow object - Gateway

- Stands for flow branching or join of branches
- Types
 - **Exclusive data-based (XOR)** 
 - Exclusive event-based 
 - Inclusive 
 - **Parallel** 
 - Complex 
- Default branch 

Gateway examples



Gateway examples II



Artifacts

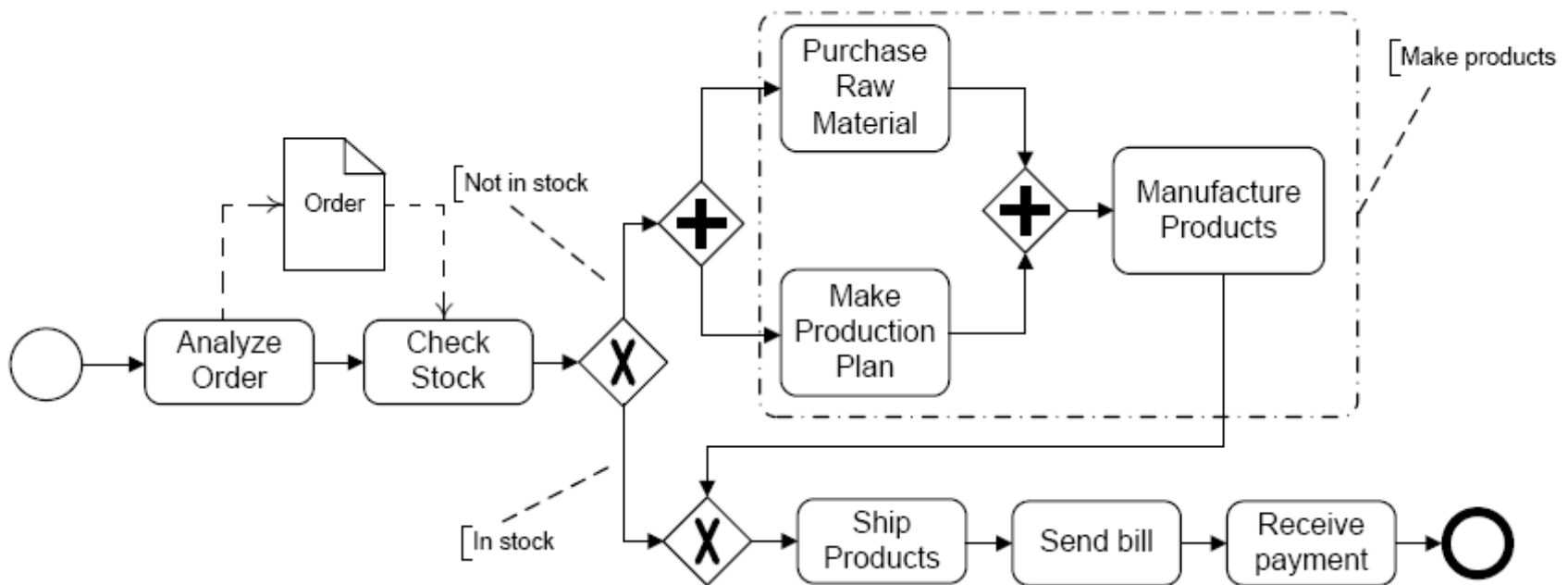
- Additional information
- Do not affect flow
- Data Objects
 - Data used in activities
 - Inputs and outputs of activities
- Annotation
 - Label, additional information
- Groups
 - Grouping of objects
(analytical/documentation reasons)



Text Annotation Allows a
Modeler to provide
additional information



Artifacts - examples

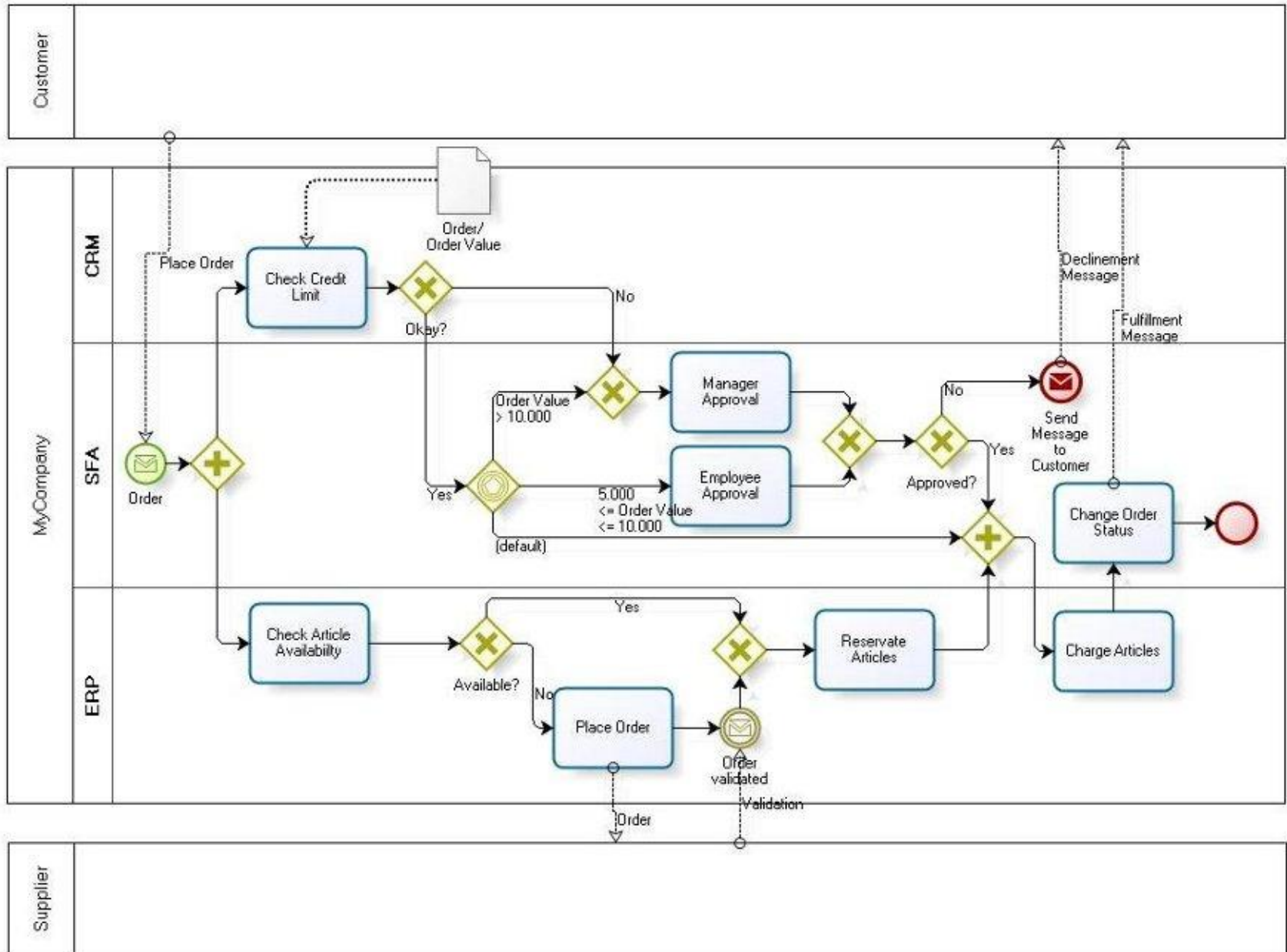


Swimlanes – Pools/Lanes

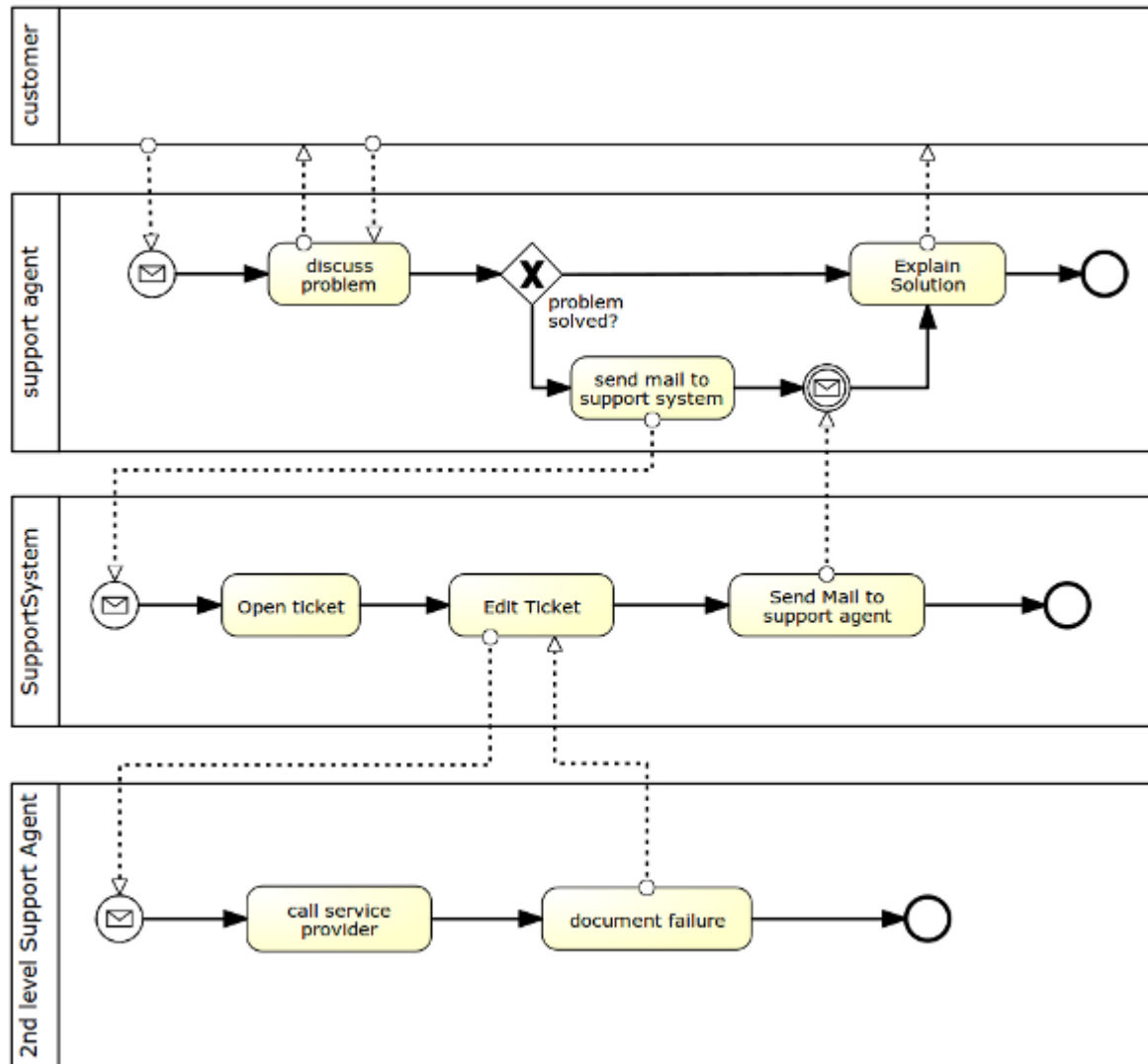
- Pool
 - Represent a participant in a process
 - Show message flows between participants
- Lane
 - Subdivision of pool
 - Express roles, departments or actors in a process



Pool/examples

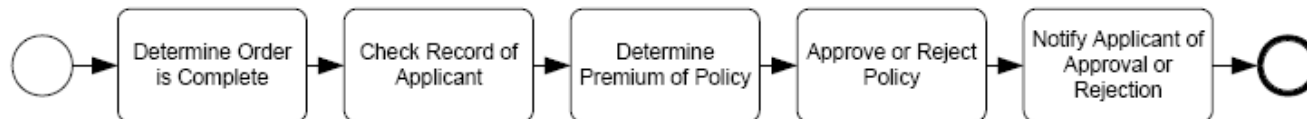


Pool/examples



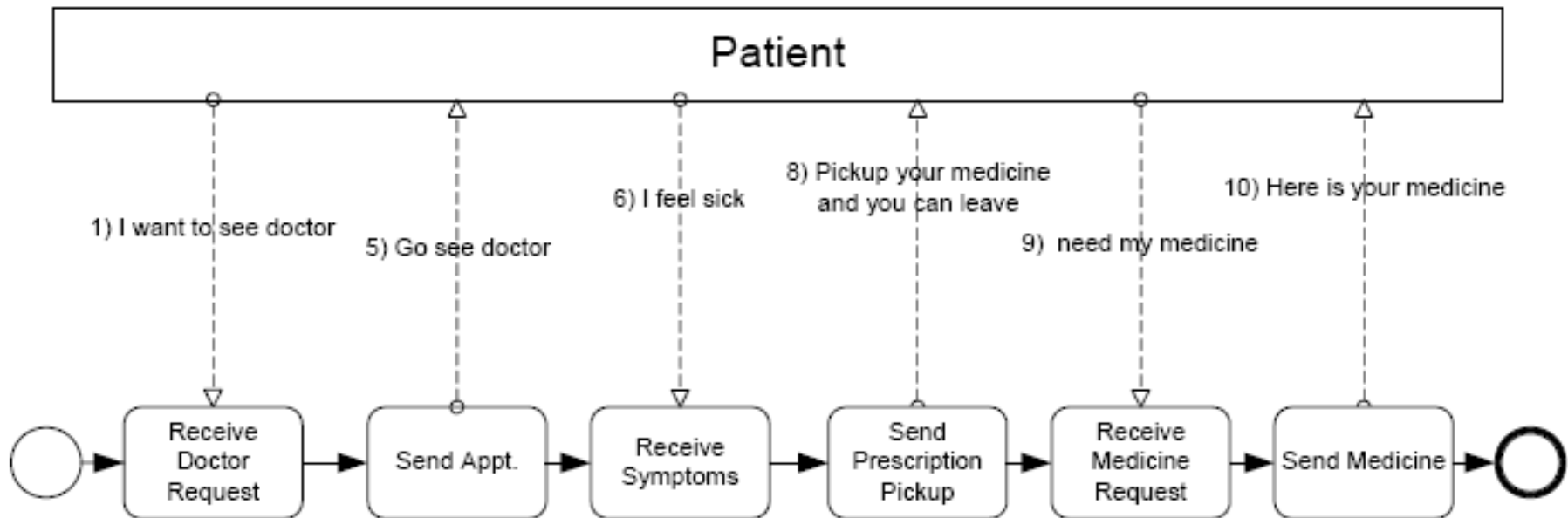
Private (Internal) Process

- From point of view of one organisation
- Activities are not visible to outside world
- One pool (the pool can be omitted)



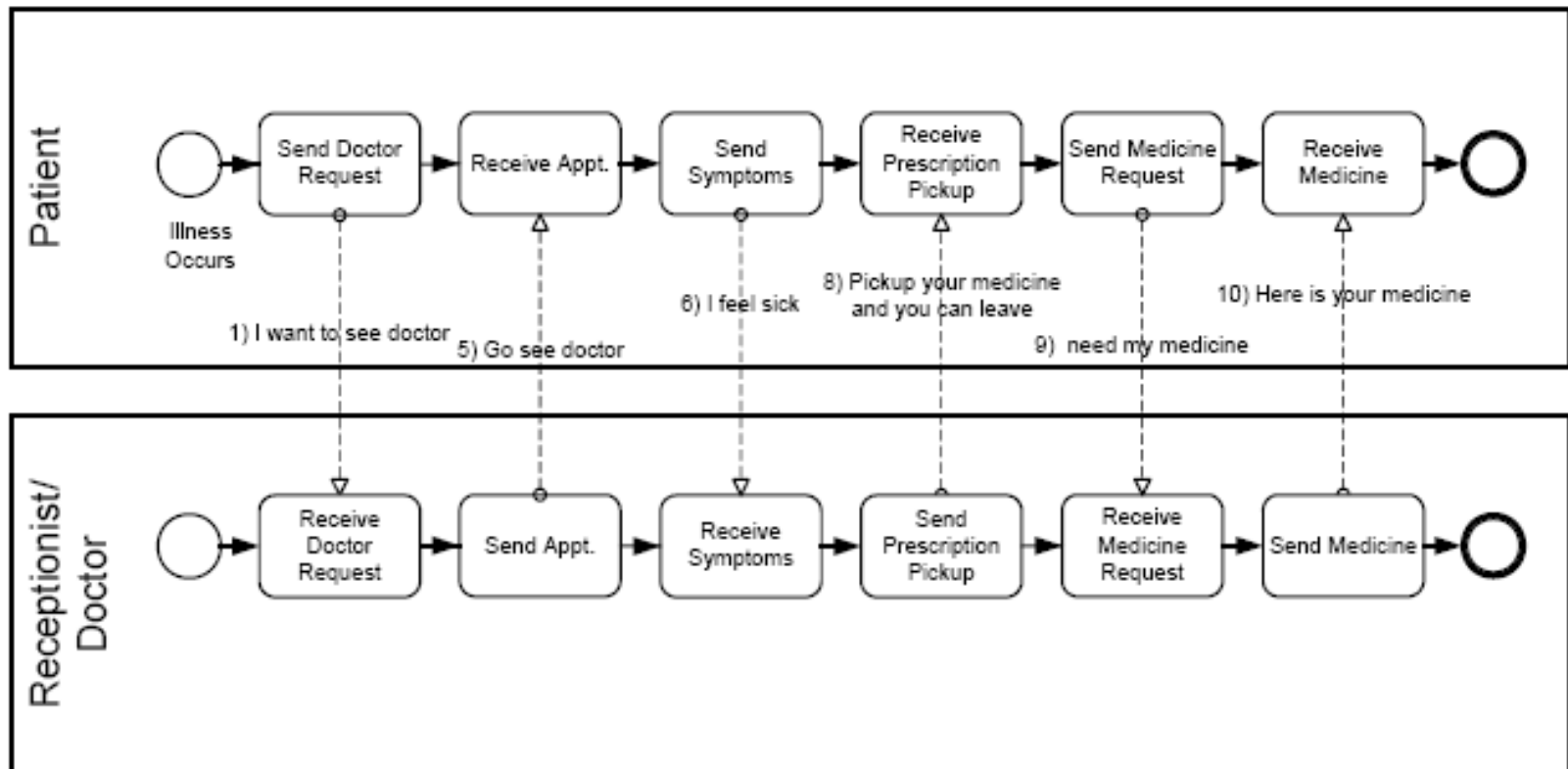
Abstract (Public) Process

- Only activities that send/receive messages
- Communication visible to outside world



Collaboration (Global) Process

- Collaboration between business entities
- Activities represent message exchange



FIN

Questions?

PV207 – Business Process Management

Spring 2012

Jiří Kolář