IN5140: Process Improvement and Agile Methods in Systems Development

Lecture 30 August 2023:

Process Modeling

UNIVERSITETET I OSLO

Yngve Lindsjørn

E-mail: ynglin@ifi.uio.no

Aspects of Process (from Intro)

- Which activities are in focus in the process?
- How much effort is spent on the various activities?
- The process also includes
  - Which and the way methods, practices, tools and techniques are used
  - Parts of the products/results of an activity
  - Roles of those involved in the process

Examples of Roles (from Intro)

- Developer
- Architect/System designer
- Graphical designer
- Documenter
- Tester
- Project manager
- User/customer representative (e.g. product owner in agile teams)

Agile Roles (from the text book chapter 5)

- Manager
  - A supporting role (not assigning tasks). “Establishing an environment that enables the team to work successfully”
- Product owner
  - Facilitates decisions about the product. Select user stories from the product backlog. Evaluates the result of the sprint (in Scrum)
- Team
  - Group of people, but also viewed as a “single” character. Self-organized. Cross-functional (teams should be formed along the lines of features)
- Members and Observers
  - The observers will give their opinion if invited, but project decisions, such as including functionality, are the privilege of members
- Customer
  - “Put the customer at the centre”. In Scrum often the Product Owner represents the customer
- Coach, Scrum Master
  - The Scrum master is responsible for making sure a Scrum team lives by the values and practises of Scrum

Examples of Tools (from Intro)

Tools for:

- Development (IDE) (e.g. Android Studio)
- Configuration/change management (e.g. GIT)
- Testing
- Diagram construction (e.g. UML, BPMN)
- Project management
- Bug & issue tracking (e.g. JIRA, GITHUB)
- Collaboration

Choice of tools is not trivial

Some Collaboration tools used in IN2000 (40 teams)

- Github
- Google hangouts
- Zoom
- Teams
- Slack
- Discord
- Facebook/Messenger
- Mozilla hubs
- Google hangouts
- Monday
- Trello
- Notion
- Google Docs, Google drive
- Invition
- Figma
- …….

UNIVERSITETET I OSLO

Project process model

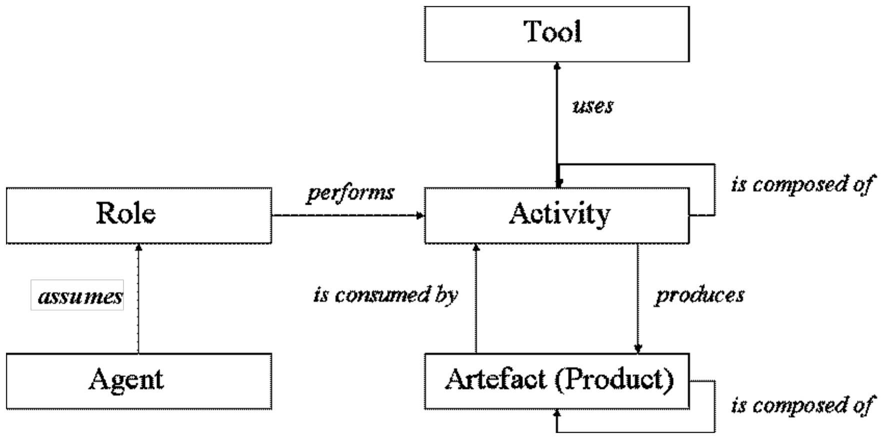
- Ideally, a process should be planned, tailored to the context and adjusted based on experience. One then needs a process model
- A process model defines **Who** is doing **What**, **When** and **How** to reach a specific goal.
  - In software engineering the goal is to build a software product or to enhance an existing one
- A good process model…
  - provides guidelines for efficient development of quality software
  - reduces risk and increases predictability
  - promotes common vision and culture

UNIVERSITETET I OSLO

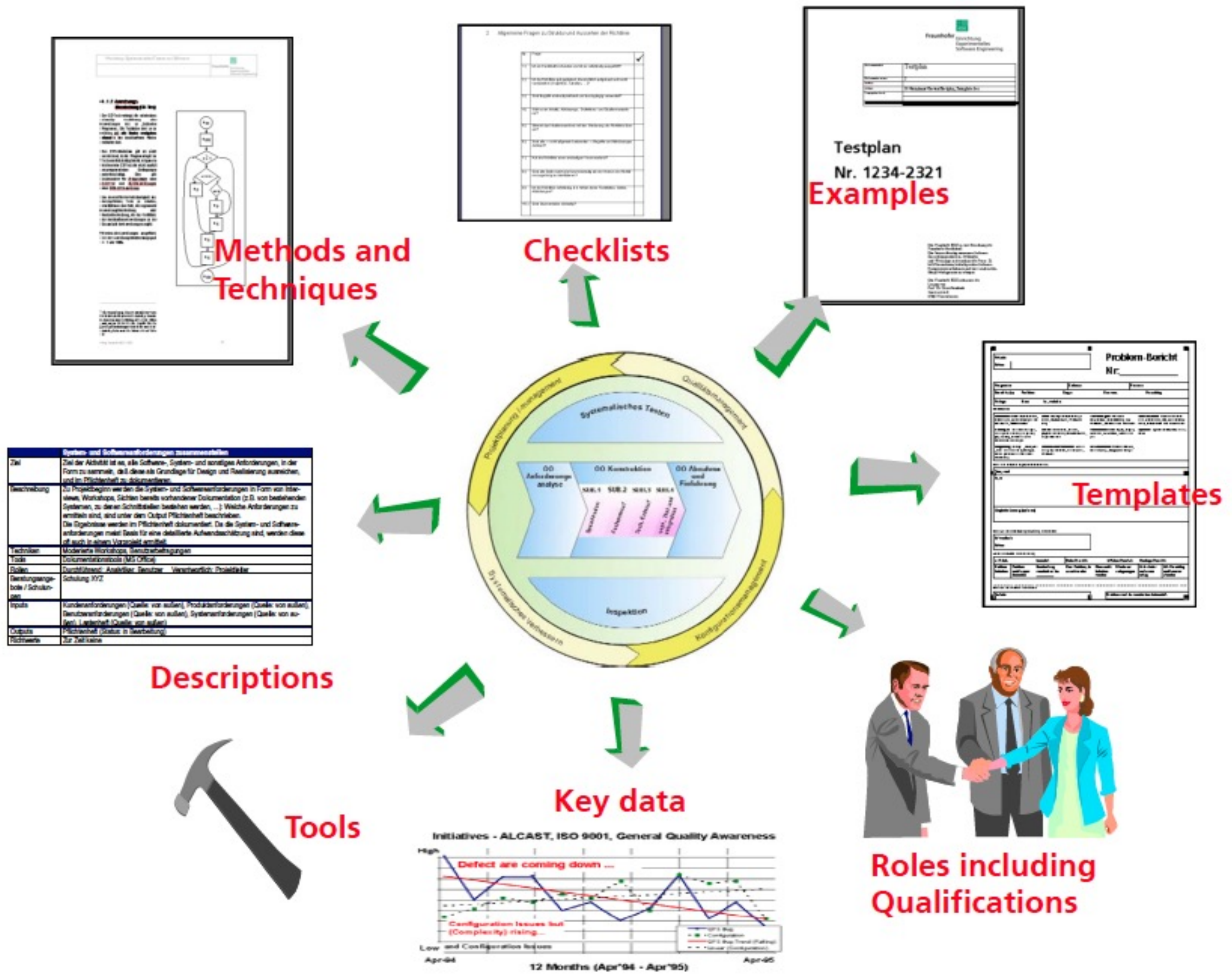
Attributes of a project process model

- The activities of the process
  - The order and scope of the activities
  - The pre- and post conditions of activities
  - Whether the activities are performed one or several times
- The roles involved in the process
- The artifacts/deliverables of the process (models, documents, source code, etc.)
- The methods, practices, tools and techniques involved in the process

Model



Typical elements of a project process model


Methods and Techniques
Checklists
Testplan
Nr. 1234-2321
Examples
Templates
Problem-Bericht
Descriptions
Systematisches Testen
Inspektion
Tools
Key data
Initiatives - ALCAST, ISO 9001, General Quality Awareness
Defect are coming down ...
Configuration issues but (Complexity) rising...
12 Months (Apr'94 - Apr'95)
Roles including Qualifications

UNIVERSITETET I OSLO

Purpose of (Descriptive) Process Modeling

- Understand the process
- Communicate (about) the process
- Support management
- Guide the work
- Identify typical deviations from the prescribed process models
- Improve software development activities
- Support measurement

UNIVERSITETET I OSLO

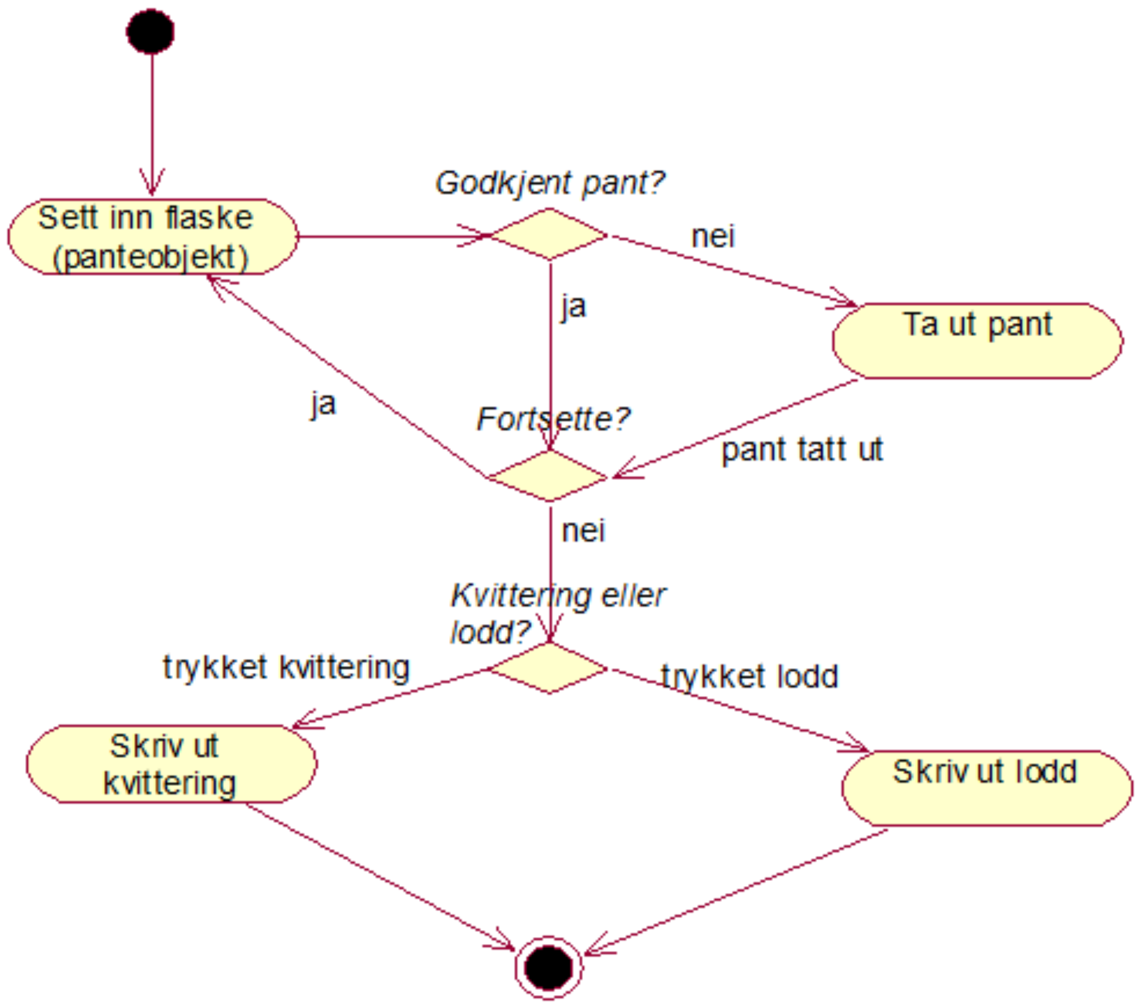
Modeling Languages and tools

- Business Process Modeling Notation (BPMN) is the most common process modeling language (www.bpmn.org/); that **is, a language for creating process models.**
- **In IN5140, we use BPMN for creating *software process models***
- Some other modelling languages for process modelling are:
- Flowcharts that can be drawn in many different tools
- IDEF which is a family of modeling languages (http://www.idef.com/)
- Activity diagrams of the Unified Modeling Language (UML)
  - **Used in INF1050 and IN1030 (Software Engineering)**

BPMN – Business Process Model and Notation

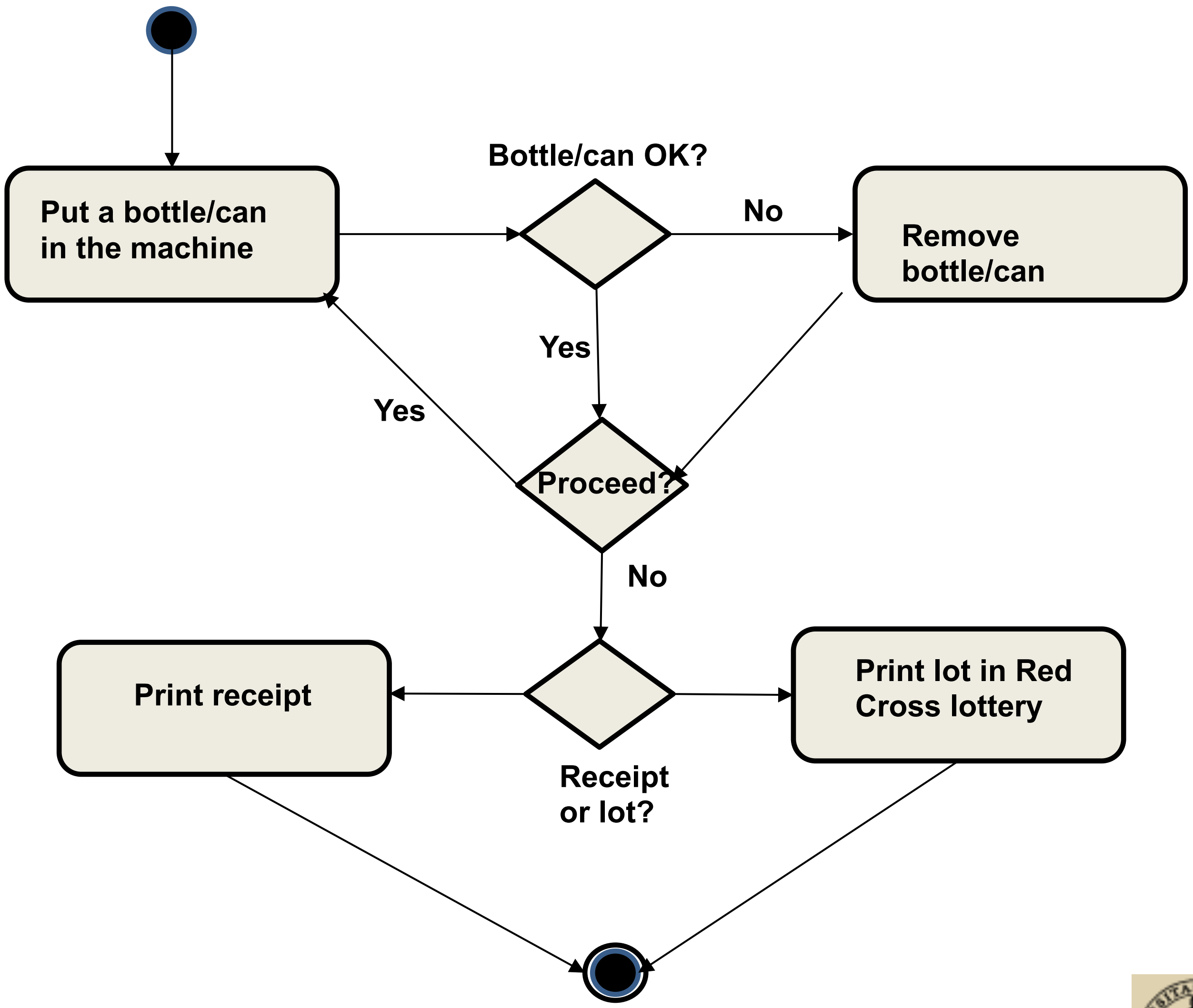
- First published in 2002 – version today is BPMN 2.0
- Official standard for Object Management Group (OMG) from 2006
- SAP, Oracle and IBM have worked with specification since 2007
- Links:
  - http://www.bpmn.org
  - http://bpmb.de/poster
- Many tools implement BPMN
  - Draw.io (easy to use)
  - Ludichart (www.ludichart.com)
  - Yaoqiang BPMN Editor is used in some of the examples in this lecture. Open Source and free for use.
    - http://sourceforge.net/projects/bpmn
  - Signavio is also used (and is much used in practise by companies etc.). Free 30 days trial. (http://signavio.com)

From INF1050/IN1030: Activity Diagram (UML) – Deliver bottles/cans in a machine (“pante flasker»)

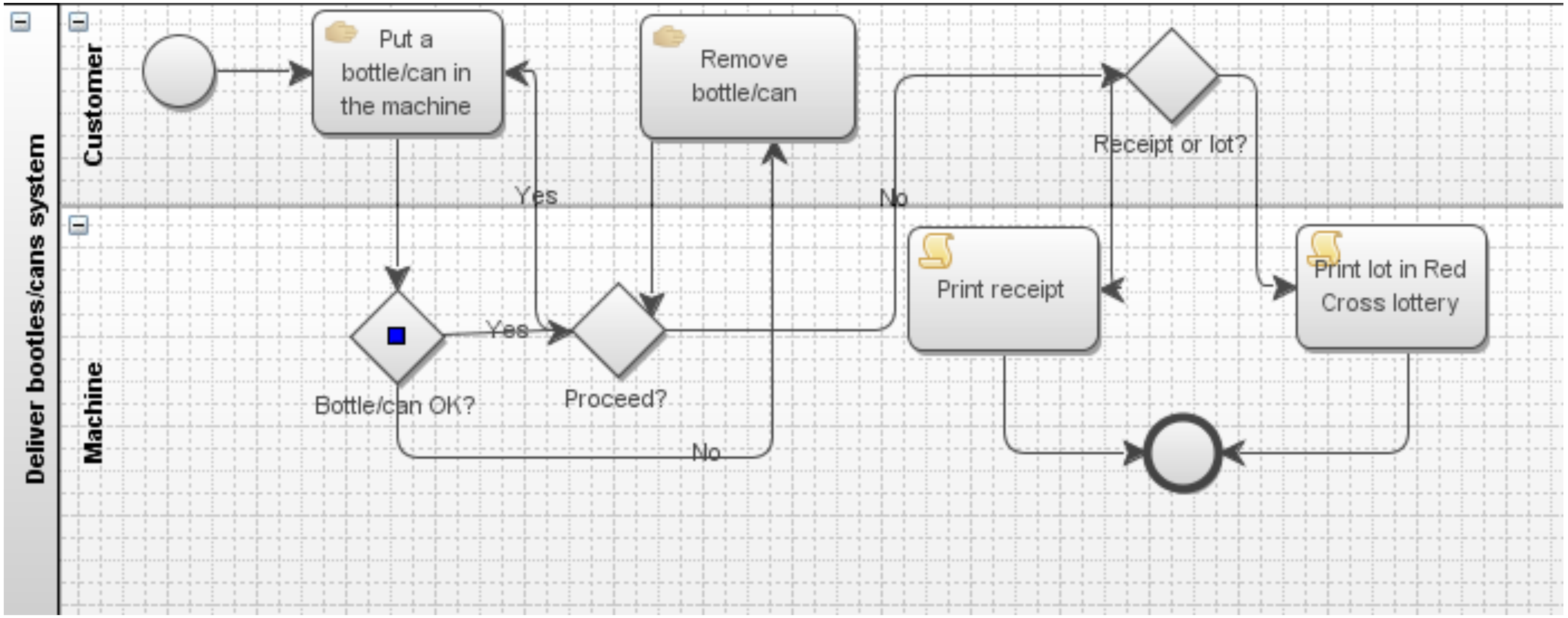


UNIVERSITETET I OSLO

From INF1050/IN1030: Activity Diagram (UML) – Deliver bottles/cans in a machine ("pante flasker»)



Deliver bottles/cans in a machine ("pante flasker») – using BPMN



UNIVERSITETET I OSLO

BPMN 2.0 - Business Process Model and Notation

http://bpmb.de/poster

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

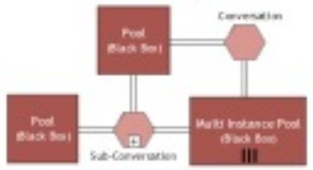
Conversations

A **Conversation** defines a set of logically related message exchanges. When marked with a ⊞ symbol it indicates a **Sub-Conversation**, a compound conversation element.

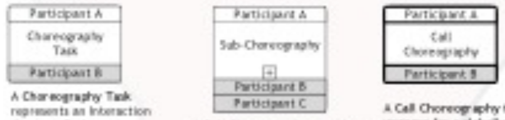
A **Call Conversation** is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Sub-conversation is marked with a ⊞ symbol.

A **Conversation Link** connects Conversations and Participants.

Conversation Diagram



Choreographies



A **Choreography Task** represents an Interaction (Message Exchange) between two Participants.

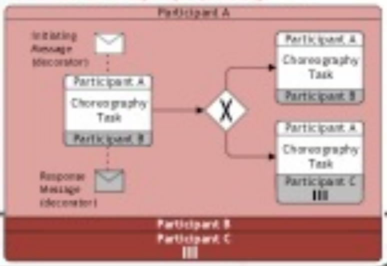
A **Sub-Choreography** contains a refined choreography with several interactions.

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.

Multiple Participants Marker denotes a set of Participants of the same kind.

Message a decorator depicting the content of the message. It can only be attached to Choreography Tasks.

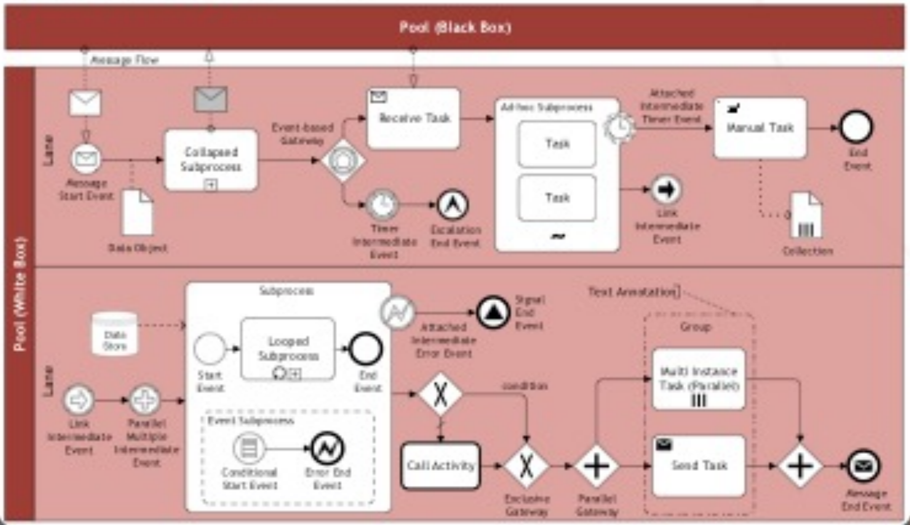
Choreography Diagram



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 timer 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.								●

Collaboration Diagram



Gateways

Exclusive Gateway	When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.
Event-based Gateway	Is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.
Parallel Gateway	When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.

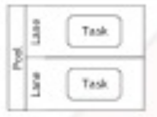
Inclusive Gateway
When splitting, one or more branches are activated. All active incoming branches must complete before merging.

Exclusive Event-based Gateway (instantiate)
Each occurrence of a subsequent event starts a new process instance.

Complex Gateway
Complex merging and branching behavior that is not captured by other gateways.

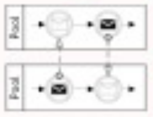
Parallel Event-based Gateway (instantiate)
The occurrence of all subsequent events starts a new process instance.

Swimlanes



Pools (Participants) and **Lanes** represent responsibilities for activities in a process. A pool or a lane can be an organization, a role, or a system. Lanes subdivide pools or other lanes hierarchically.

Message Flow symbolizes information flow across organizational boundaries. Message flow can be attached to pools, activities, or message events. The Message Flow can be decorated with an envelope depicting the content of the message.



The order of message exchanges can be specified by combining message flow and sequence flow.

BPM OFFENSIVE BERLIN

HPI Hasso Plattner Institut

camunda

inubit

SIGNAVIO

© 2011

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.

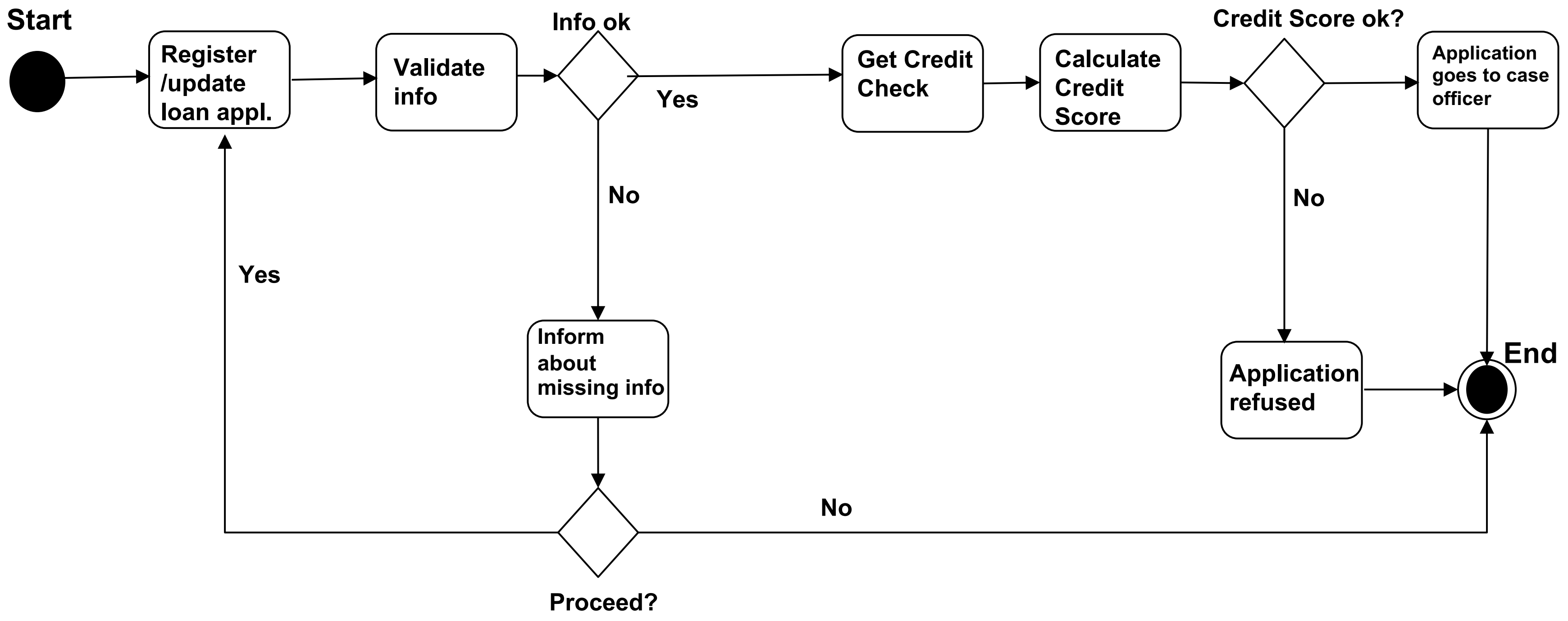
A **Data Input** is an external input for the entire process. A kind of input parameter.

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.

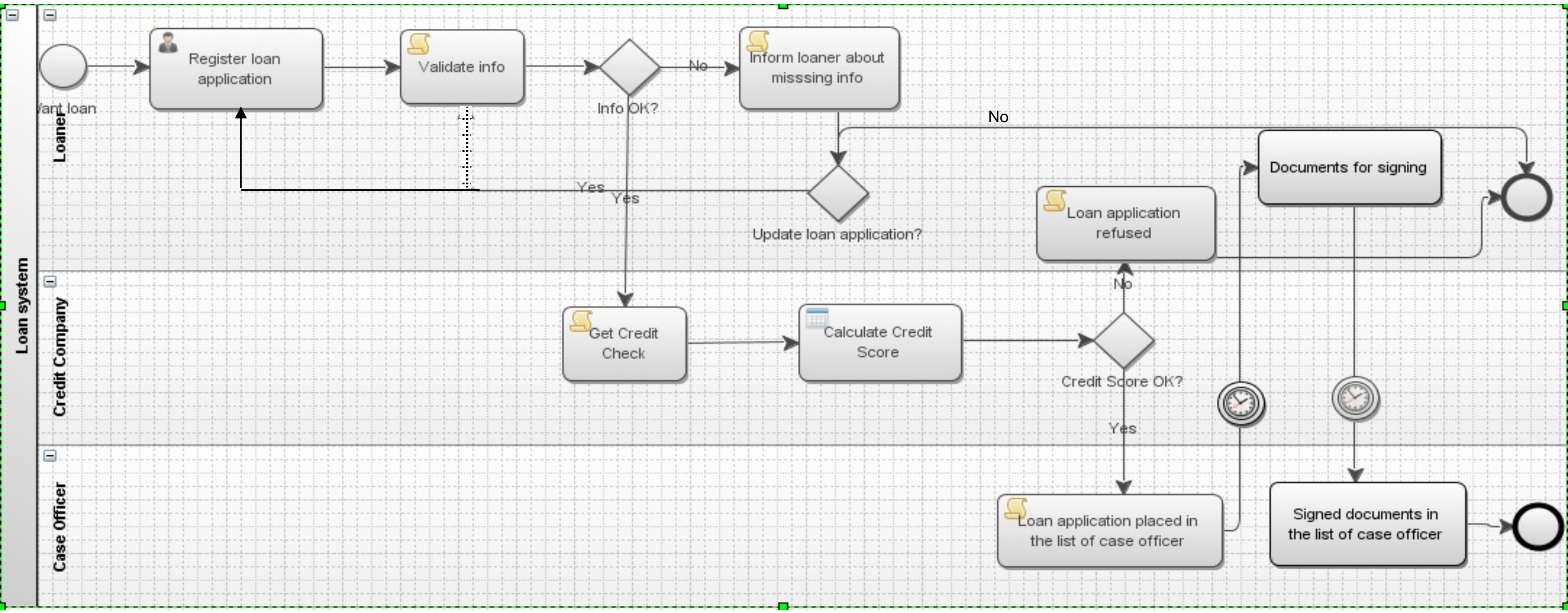
From INF1050/IN1030: Activity diagram “Registrer lånesøknad” (register loan application)



Exercise

- Discuss and make a BPMN diagram of the «Register loan» usecase. Use the actors Loaner (User), Credit company and case officer (lånekonsulent). Extend the task so that documents are sent for signing.

Example BPMN – Register a loan application



UNIVERSITETET I OSLO

Example of process modelling – eating dinner

Suppose you are in a restaurant. You have found a table and want to have dinner:

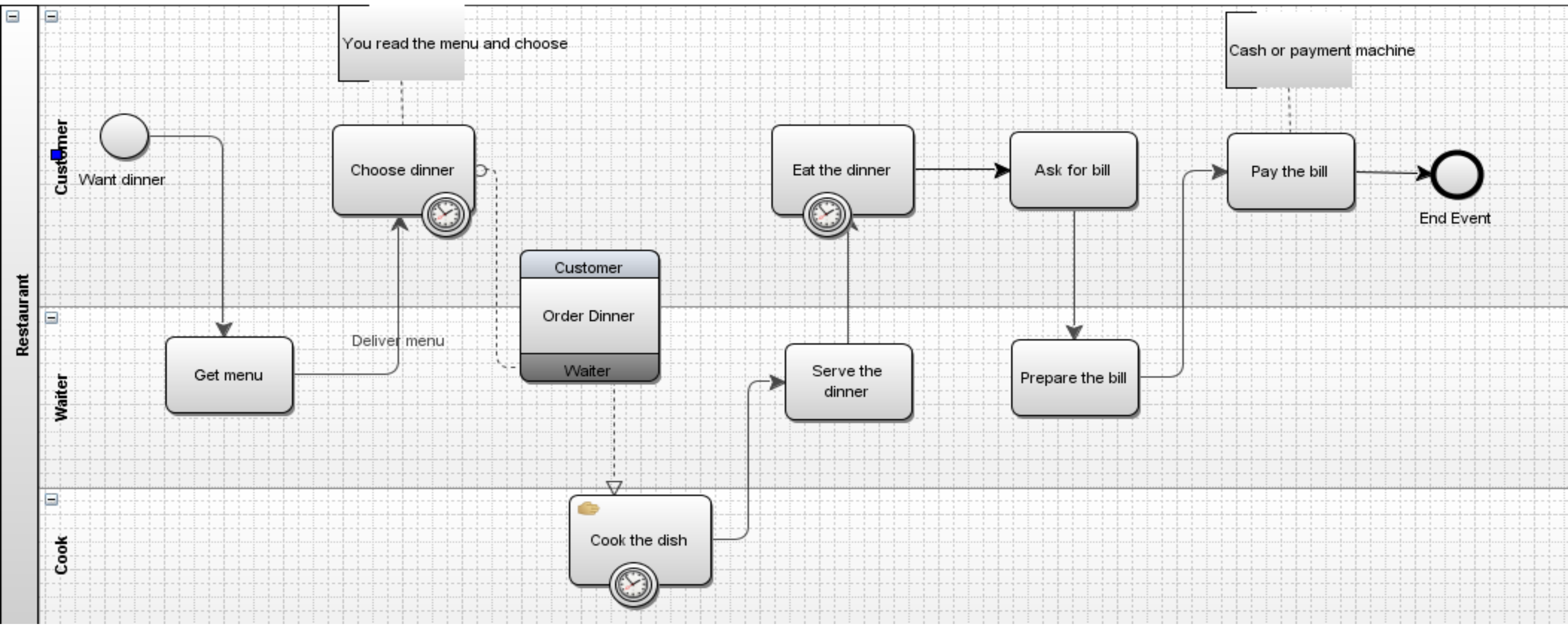
> You get a menu from a waiter, choose a dish and make an order. A waiter writes it down, goes to the kitchen and requests the cook to make your dish. When the dish is ready, the cook informs waiter and waiter serves the dish to you. You eat the meal and pay for it.

Exercise

- Discuss and make a BPMN diagram of the «Eating dinner» usecase. Call the pool Restaurant and use the actors Customer, Waiter and Cook.

UNIVERSITETET I OSLO

Example – Eating dinner



Exercise in group session

- Read the textual description of the process of developing a new functionality for a web application.
- Use BPMN and create a process model of the process of developing a new functionality for the existing web application.

Exercise – INF5140: Process Modeling using BPMN

Background

- Company X uses a web application to support its business. The web application was developed and is currently maintained by Company Y. The two companies work together on improvements and further development; X identifies needs for new functionality and orders this from Y, and Y regularly delivers new versions of the web application with the requested functionality to X.

The process of developing new functionality

- Company X has a Product owner for the web application who is responsible for identifying new business opportunities, for describing new requirements and for testing. The Product owner regularly hands over a requirement specification to the Delivery manager at Company Y. The Delivery manager is responsible for the successful delivery of new versions of the application. The Delivery manager cooperates with a team of developers in company Y. They discuss the requirements from company X and estimate the tasks. Then the Delivery manager informs the Product owner about the estimates and costs. Based on this information, the Product owner decides which requirements should be prioritized for the new version of the system and communicates this back to the Delivery manager. Then the developers start development. When the new features have been developed, they are deployed to the test environment by the Operation department at Company Y. The Delivery manager requests the Product owner to test that implemented features correspond to the specified requirements. If some of the features do not pass the test, the Product owner makes a list of errors and informs the Delivery manager, who in turn informs the developers of the necessary changes and error corrections. When all tests are passed, the test cases are documented and sent to the Delivery manager, who requests the Operation department to deploy the new version to the production environment. The Operation department deploys the new version.