MUNI
FI

PB007 Week 04

Samuel Sabo

Activity Diagram

- Object oriented model of a chosen **activity** flow
- The Activities usually represent:
  - use cases
  - operations and methods
  - algorithms
  - business processes

Basic components

Activity – to be decomposed (top-level rectangle)

Nodes (shapes inside)

- **action** nodes – atomic unit of work within an activity
- **control** nodes – controlling flow through the activity
- **object** nodes – objects used in the activity

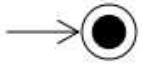
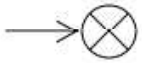
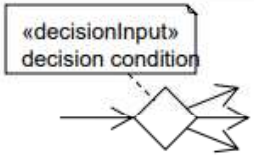
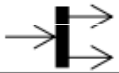
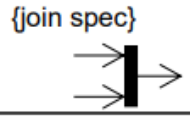
Flows/Edges (arrows)

- **control** flow
- **object** flow

Swimlines – separate logical parts (lines dividing rectangle)

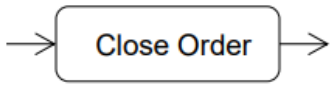
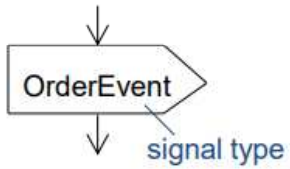
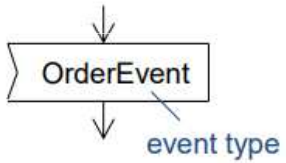
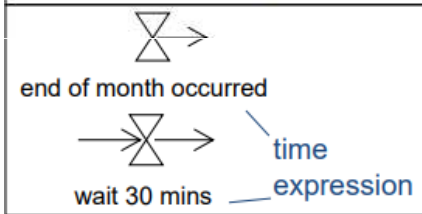
MUNI FI

Control nodes

control node syntax	control node semantics	
	Initial node – indicates where the flow starts when an activity is invoked	
	Activity final node – terminates an activity	Final nodes
	Flow final node – terminates a specific flow within an activity. The other flows are unaffected	
	Decision node– guard conditions on the output edges select one of them for traversal May optionally have inputs defined by a «decisionInput»	
	Merge node – allows through any of its input edges	
	Fork node – splits the flow into multiple concurrent flows	
	Join node – synchronizes multiple concurrent flows May optionally have a join specification to modify its semantics	

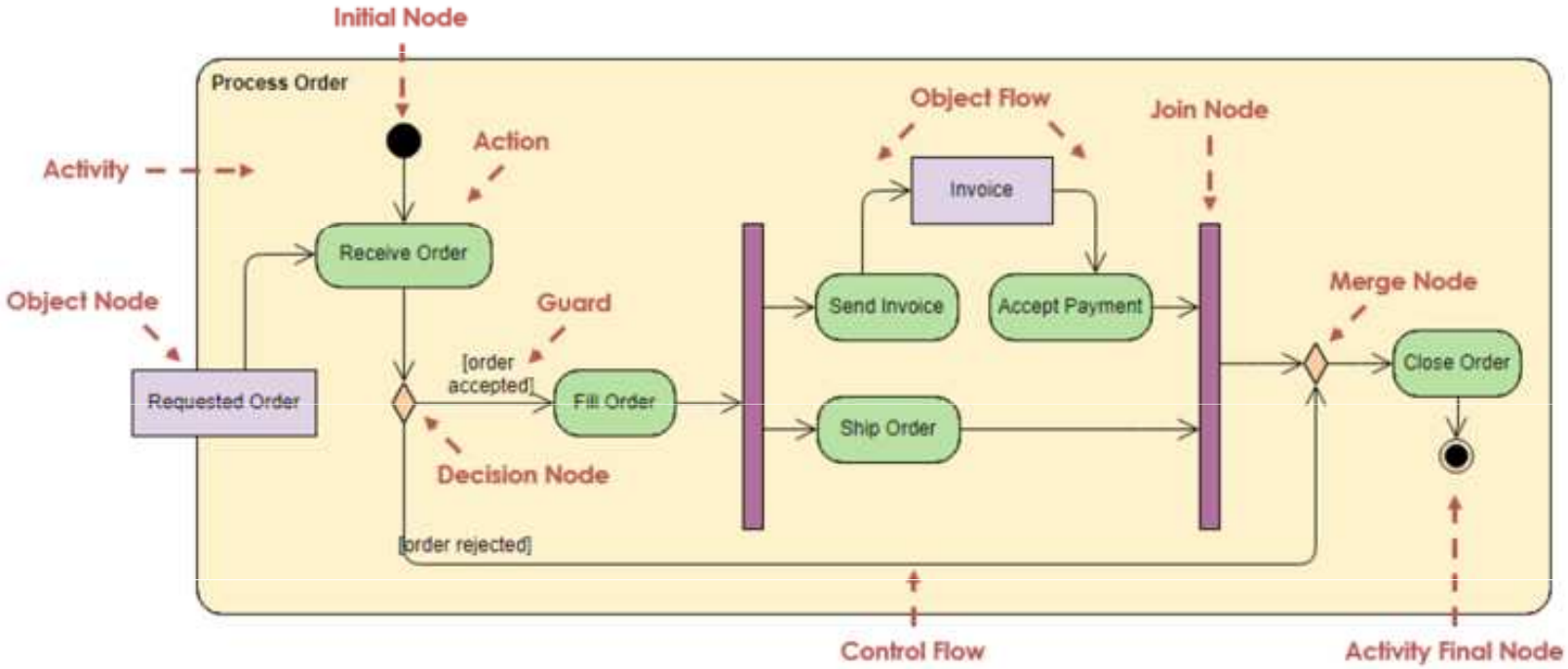
MUNI FI

Action nodes

action node syntax	action node semantics
	Call action - invokes an activity, a behavior or an operation. The most common type of action node. See next slide for details.
	Send signal action - sends a signal asynchronously. The sender *does not* wait for confirmation of signal receipt. It may accept input parameters to create the signal
	Accept event action - waits for events detected by its owning object and offers the event on its output edge. Is enabled when it gets a token on its input edge. If there is *no* input edge it starts when its containing activity starts and is *always* enabled.
	Accept time event action - waits for a set amount of time. Generates time events according to it's time expression.

MUNI FI

Example

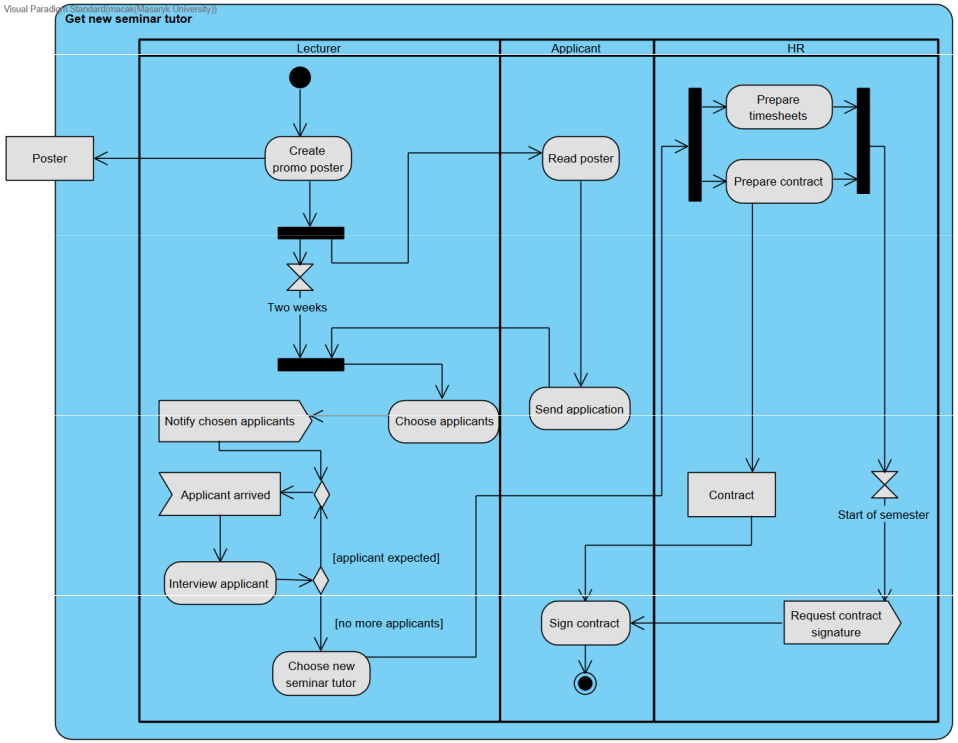


MUNI FI

Swimlines

- Logical grouping of related actions (e.g. based on actors, use cases, ...)
- They usually represent individual uses cases connected with <<include>> or <<extend>>

Example


Get new seminar tutor
Lecturer
Applicant
HR
Poster
Create promo poster
Read poster
Prepare timesheets
Prepare contract
Two weeks
Notify chosen applicants
Choose applicants
Send application
Applicant arrived
Contract
Start of semester
[applicant expected]
Interview applicant
[no more applicants]
Sign contract
Request contract signature
Choose new seminar tutor

Task for this week

- Review the advanced use case diagram and textual specifications from the previous session. Fix any problem.
- Choose 2 use cases and create activity diagrams for them
  - mark them on the diagram
- Submit this week report in homework vault week04 in format **surname1-surname2-surname3.pdf**
- If you have not yet set up you report generation in VP refer to week02