

UML Activity Diagrams

Vreda Pieterse and Linda Marshall

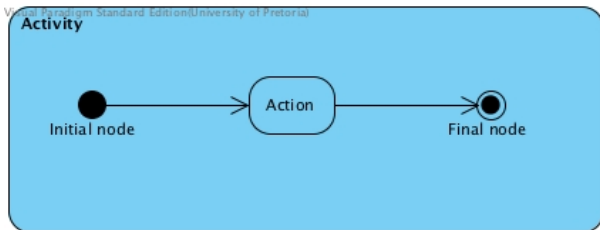
Department of Computer Science
University of Pretoria

12 September 2014

- Model complicated workflows in operations on objects
- Model complex activities in more detail

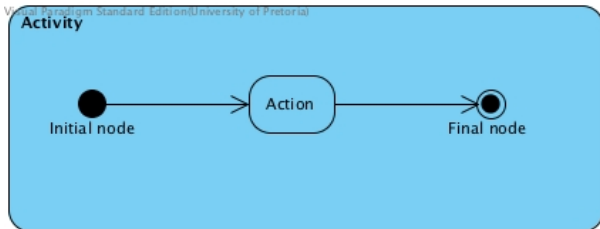
- Action Nodes
- Activity Edges
- Control Nodes
- Swimlanes

Initial Node



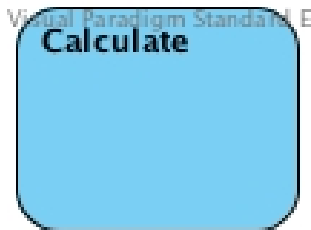
- Starting point
- An activity diagram may at most have one initial node

Final Node



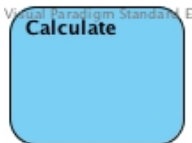
- Point where activity terminates
- An activity diagram may have many final nodes

Action Node



- Describe an action
- Represent a function (method)
- May contain sub-activities (infinitely)

Activity



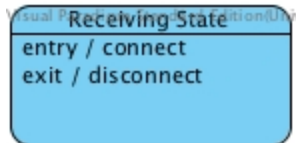
actions

fire immediately

guards on decision only

actions are actions

State



modes

fire on an event

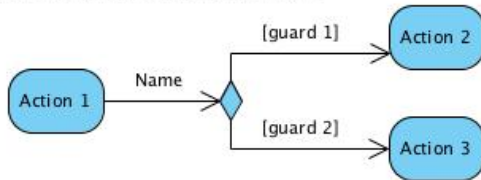
guards on all

actions on transitions

actions in states

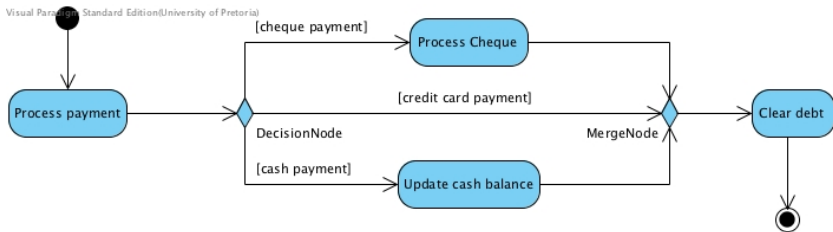
Activity Edge

Visual Paradigm Standard Edition(University of Pretoria)



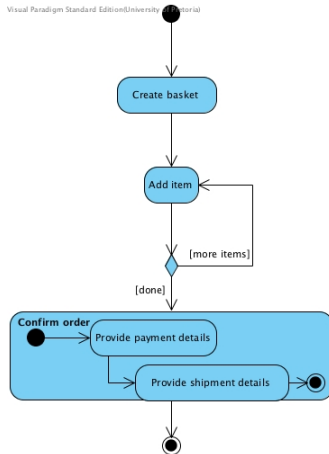
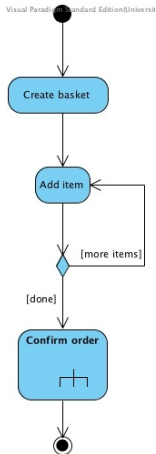
- Indicate flow between actions
- May have a name (plain text)
- Guards only for decisions (text in [])
- May not have actions (text after /)

Decision and Merge

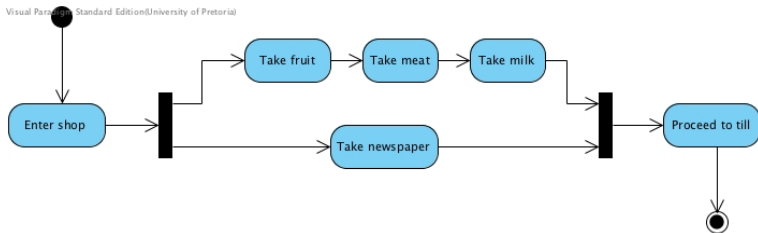


- Alternate flows
- Needs guard conditions for alternatives

Composite Activities

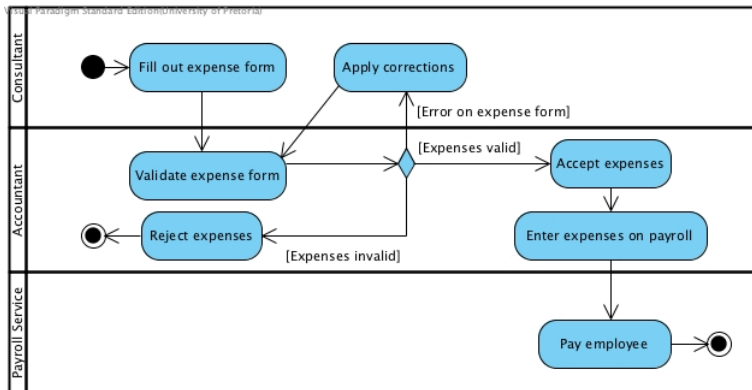


Fork and Join (Parallel flows)



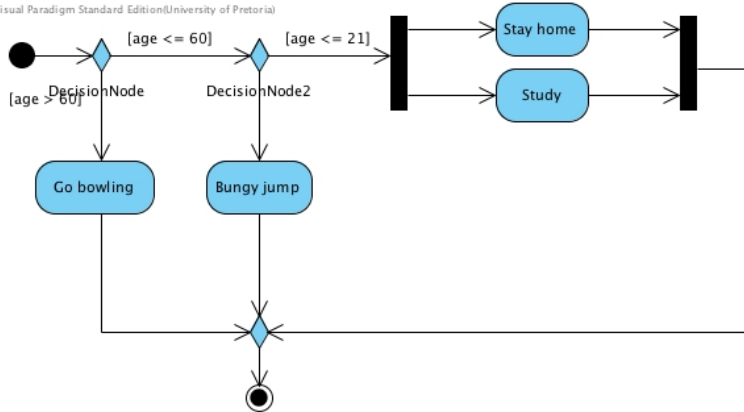
- Both paths are executed at the same time by different threads
- Join fires only after all flows have reached it. Synchronisation point

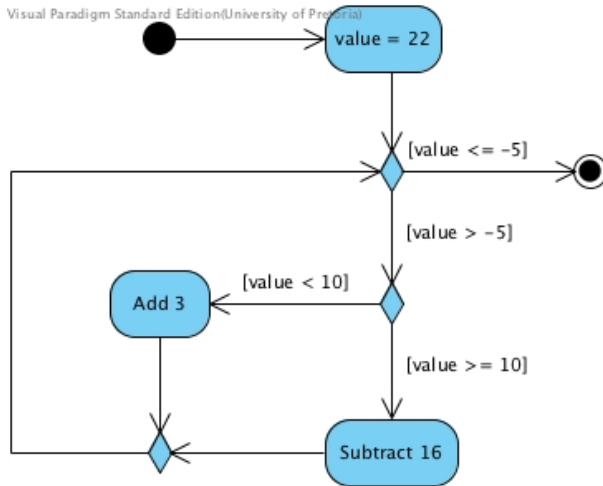
Swimlanes to separate responsibilities in different classes



```
if (age > 60){  
    goBowling(); }  
else if (age > 21){  
    bungyJump(); }  
else{  
    stayHomeThread.join();  
    studyThread.join();  
    stayHomeThread.detach();  
    studyThread.detach(); }  
}
```

Visual Paradigm Standard Edition(University of Pretoria)





```
int value = 22;  
while (value > -5){  
    if (value < 10){  
        value += 3;  
    }  
    else{  
        value -= 16;  
    }  
}
```