Modelling Dynamics (2): Activity Diagrams

CS3 / SEOC1

Note 11

Activity Diagrams

State Diagrams describe the state change behaviour of the classes involved in the system. By contrast, Activity Diagrams *are intended to describe how activities are coordinated to provide a service. The service can be at different levels of abstraction. For example, a service might be an individual operation or some higher level service (see later).*

Activity Diagrams

- In the UML semantics these are reducible to State Diagrams with some additional annotations.
- The additional annotations are intended to capture how activities are coordinated. In particular, it is easily possible to represent concurrency and coordination in Activity Diagrams.

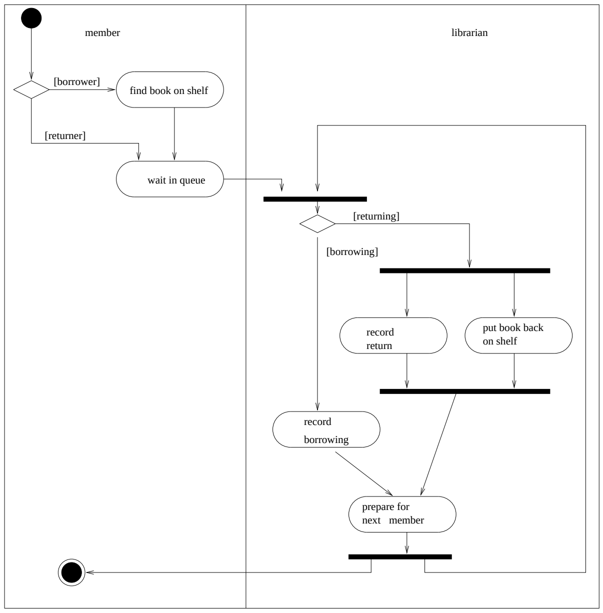
Activity Diagrams (2)

- Activity diagrams can be used to describe:
  - The events needed to achieve some operation, particularly where the operation is intended to achieve a number of different things that require coordination.
  - How the events in a single use case relate to one another. In particular, use cases where activities may overlap and require coordination.
  - How a collection of use cases coordinate to create a *workflow* for an organisation.

Activity vs State Diagrams

- In State Diagrams the vertices represent states of an object in a class and edges represent occurrences of events.
- In Activity Diagrams the vertices represent the carrying out of an activity and the edges represent the transition on the completion of one collection of activities to the commencement of a new collection of activities.
- Activity Diagrams take a "higher level" view of activities.

Activity Diagram for Library
Using UML: Chapter 11



Simple Activity Diagrams

activity: these are the vertices of the diagram. This is like a state where the criterion for leaving the state is the completion of the activity.

transition: unlabelled arc from one activity to the next. Transitions take place when one activity is complete and the next can commence.

synchronisation bar: represented as a horizontal bar in the diagram. A synchronisation may have many in-arcs from activities and a number of out-arcs to activities. The bar represents synchronisation of the completion of those activities with arcs into the transition. On an occurrence of the transition all the activities with arcs from the transition are initiated.

Simple Activity Diagrams (2)

decision diamond: Indicates where a choice should be taken.

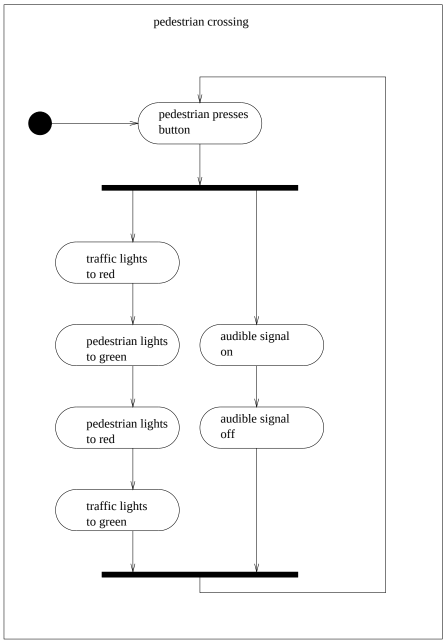
start and stop markers: mark initial and final activities.

mal-formed diagrams: it is possible to form ill-formed diagrams that require multiple activations of activities or can allow deadlock.

Partitioning into “Swimlanes”

- It is possible to partition the activity diagram so that parts of it appear in the column relevant to that element in the partition. Each column of the partition is called a “swimlane”.
- For example partitions may be constructed on the basis of:
  - the class or actor doing the activity.
  - the use case the activity belongs to.
- It may be helpful to have more than one partitioning of the same activity.
- Partitioning by use case can help document how use cases interact.
- Partitioning by class and actor can help to identify new associations that have not been documented in the Class Model.

Activity Diagram for Pedestrian Crossing

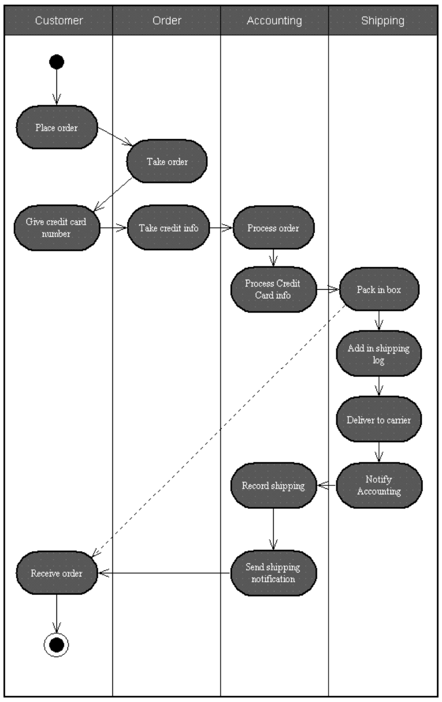


Summary

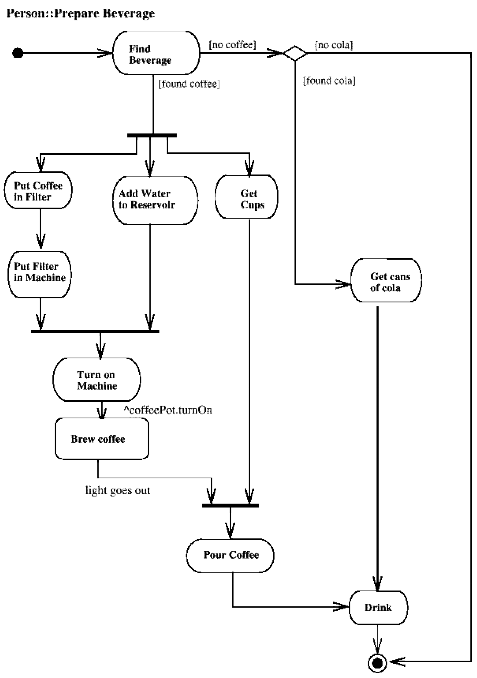
- Activity Diagrams are good for describing synchronisation and concurrency between activities.
- Activity diagrams are useful for capturing detailed activities but can also capture elements of the high level workflow the system is intended to support.
- Partitioning can be helpful in investigating responsibilities for interactions and associations between objects and actors.

Example Activity Diagram (1)

UML Activity Diagram: Order Processing

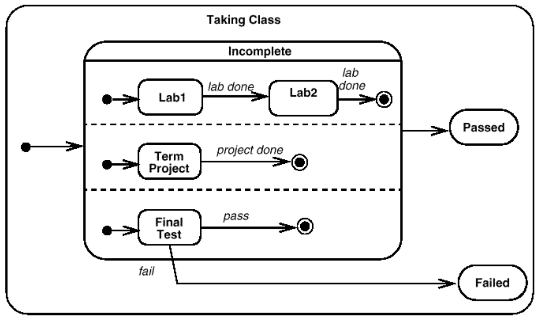


Example Activity Diagram (2)

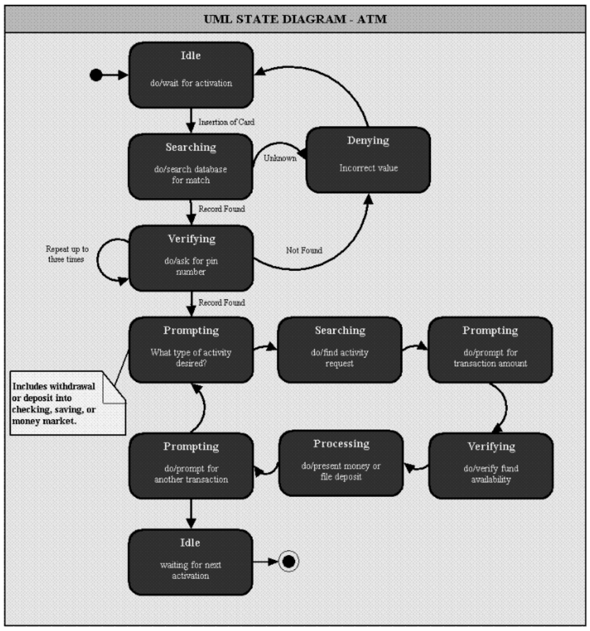


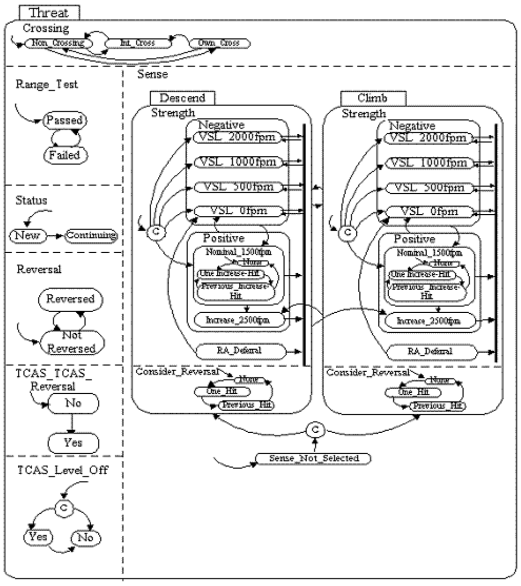
Example State Diagram (1)

Figure 44. Concurrent substates



Example State Diagram (2)





TACAS State Diagram