### Hierarchical State Machines

UML distinguishes the following kinds of states:

- **Simple state**
  - entry
  - do
  - exit
  - entry

- **Final state**
- **Composite state**

**Hierarchical states** can be connected through:

- **OR**
- **AND**

**Submachine states** can be used to define

- **(shallow) history**
- **Deep history**
- **Fork/join**, junction, choice
- **Entry point**
- **Exit point**
- **Terminate**

---

### Exercise 4 (i)

1. `(u_1, e_1)`
2. `(u_2, e_2)`
3. `(u_3, e_3)`
4. `(u_4, e_4)`
Passive and Reactive / Rhapsody Style: Example

What about non-active objects?

Active and Passive Objects

– 17 – 2017-01-17 – Sactpass –
• Rhapsody also supports non-active objects—their instances share an event pool with an active object.

• Behavioural Features: exist.

• Semantic Variation Points are legion—but manageable, e.g. by appropriate modelling guidelines (stick to "the beaten track").

• Interactions can be used for reflective descriptions of behaviour, i.e.
  • describe what behaviour is (un)desired, without (yet) defining how to realise it.

• One visual formalism for interactions:
  • Live Sequence Charts
  • partially ordered locations,
  • instantaneous and asynchronous messages,
  • conditions and local invariants

Later: pre-charts.

References


