Statecharts
- Introduced in mid '80s by David Harel
- Specifying reactive systems; system characterized by being event-driven, continually adapting to external and internal stimuli.
- Very popular formal technique
  - Statecharts
- Modifications, Adaptations, Extensions exist
  - UML
  - C

- Added hierarchy, concurrency, and communication

- Goals: Small diagrams can express complex behavior

![Statechart Diagram]

\( (P) \) = transition from A to C if event y occurs in state A and condition P holds at the instance of occurrence

XOR: either A or C, not both
- Initial states (default arrows)

- A is default
- A within D is default
- D is default away from B and C
- A+B, A is default within D (among A and C)

- Example: Wrist watch

- Display time: (AM/PM or 24 hour)
- Date: day, month, day of week
- Chime: beep on hour
- Two alarms
- Stop watch
- Light for illumination
- Weak battery indicator
- Beep sound
FSM: Hierarchy and Concurrency

Add Hierarchy

Concurrency

You could implement within concurrency
Concurrent / Independence

- Initial state: \((A, B, C, D, E, F)\)

- Synchronization: event \(a\) will cause \(B \rightarrow C\) and \(F \rightarrow G\)
  \[ (A, B, C, D, E, F) \rightarrow (C, G) \]

- Independence: event \(a\) will cause \(E \rightarrow E\) but has no effect in \(A\)

- Dependence: \((E \rightarrow G)\)

- Diagram:

```
\[ \text{Diagram showing states and transitions} \]
```
- Actions: Instantaneous occurrences that take zero time
  - event of output from system
  - can use both Moore and Mealy semantics

- Activities: outputs of events that require some time
  - by shut(a), keep (t)
  - by new condition active (t)
  - by throughout (x) - equivalent to shut(a) at entry and keep(t) at exit.