CS 320
Fundamentals of Software Engineering

Lecture 10: System Modeling (3) — Sequence Diagram & State Diagram
Midterm 1

- Oct. 2nd, in class (1:25pm - 2:40pm)
- 75 mins
- Close book
- VSCI 12
- 10% of the final grade
- Covers materials up to today’s lecture
Question Types

- T/F with justification
- Multiple choice (one correct answer)
- Short answer
T/F

- Use-case actors are always people, never system devices

- False. Actors can be either people, external systems, or hardware devices.
Choice

* Which of the following should be included in an SRS document.

A. Class definition

B. Methods and attributes of classes

C. Activity diagrams

D. Project scope
Choice

Which of the following is NOT an objective of having a requirement specification?

A. Define set of software requirements that can be validated

B. Describe customer requirements

C. Develop an abbreviated solution for the problem

D. Establish basis for software design
Short Answer

* At what circumstances the waterfall model is suitable?
  * When the requirements are well defined and well understood.
  * When changes are limited.
Design a simple software system for ATM machines, and draw the use case diagrams of your design.
Behavioral Models

• Behavioral models are models of the dynamic behavior of a system as it is executing.

• Behavioral models show what happens when a system responds to a stimulus from its environment.

• You can think of these stimuli as being of two types:
  • Data (sequence diagram)
  • Events (state diagram)
Sequence Diagrams

• Sequence diagrams are used to model the interactions between the actors and the objects within the system

• A sequence diagram shows the sequence of interactions that take place during a particular use case
Sequence Diagram

* View patient information
Sequence Diagram

* Transfer data
Sequence Diagram

* Order processing
Data-driven Systems

* Sequence diagrams are suitable for data-driven systems

* Data-driven systems are primarily driven by data. They are controlled by the data input to the system, with relatively little external event processing
Event-driven Systems

- Real-time systems are often event-driven, with minimum data processing
- Event-driven modeling shows how a system responds to external and internal events
- It is based on the assumption that a system has a finite number of states and that events may cause a transition from one state to another (state diagram)
State Diagram
State Entry and Exit Action

LampOn

entry/lamp.on();
exit/lamp.off();

e1

e2
Order of Actions

Resulting action sequence:

```c
printf("exiting");
printf("to off");
lamp.off();
```

---

**LampOn**

- entry/lamp.on();
- exit/printf("exiting");

---

**LampOff**

- entry/lamp.off();
- exit/printf("exiting");
- off/printf("needless");
- printf("exiting");
- printf("needless");
- lamp.off();
Internal Transitions

- Self-transitions that bypass entry and exit actions

**LampOff**

- entry/lamp.off();
- exit/printf("exiting");
- off/null;

Internal transition triggered by an “off” event
State (“Do”) Activities

* Forks a concurrent thread that executes until
  * the action completes or
  * the state is exited through an outgoing transition

```c
entry/printf("error!")
  do/while (true) alarm.ring();
```

“do” activity
Guards

- Conditional execution of transitions
Dynamic Conditional Branching

- Guards are evaluated at the instant when the decision point is reached

```
bid / gain := calculatePotentialGain(value)
```

```plaintext
Selling

- [gain >= 100] / sell
- [gain < 100] / reject

Dynamic Choice point

Unhappy

- [(gain >= 100) & (gain < 200)] / sell

Happy
```
Hierarchical State

- States decomposed into state machines

![State Machine Diagram]

- LampOff
  - entry/lamp.off()
- off/
- LampOn
  - entry/lamp.on()
- FlashOn
  - entry/lamp.on()
- FlashOff
  - entry/lamp.off()
- LampFlashing
  - flash/
  - 1sec/
  - on/
  - on/
  - on/
“Stub” Notation

- Notational shortcut: no semantic significance
High-level transitions

Group Transitions
Trigger Rules

* Two or more transitions may have the same event trigger
* innermost transition takes precedence
* event is discarded whether or not it triggers a transition
The States of a System

- **State** — a set of observable circumstances that characterizes the behavior of a system at a given time

- **State transition** — the movement from one state to another

- **Event** — an occurrence that causes the system to exhibit some predictable form of behavior

- **Action** — process that occurs as a consequence of making a transition
Behavioral Modeling

* Make a list of the different states of a system (How does the system behave?)

* Indicate how the system makes a transition from one state to another (How does the system change state?)
  
  * indicate event
  
  * indicate action

* Draw a state diagram or a sequence diagram
State Diagram

- Microwave oven
Microwave oven operation