

CMPE 276 Software Engineering

Lecture 4 UML

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

1

Projects

- Paper repository
 - Jason Rohrer, Ian Brown
- Delta-V
 - Sung Kim, Guozheng Ge, Kai Pan
- Servlet configuration manager
 - Rita Garcia
- Program for motion control in animation
 - Mark Slater
- Lego Mindstorm / Pocket PC networking
 - Venkatesh Rajendran, Vaibhav Bhandari

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

2

Assignment - due Oct 10

- Everybody:
 - Read Harel: *Statecharts: A Visual Formalism for Complex Systems*
 - Read B.P Douglass: *UML Statecharts*
(Both available on the class web page)

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

3

Assignment - due Oct 10

- Tools track:
 - Write use cases and UML Statecharts for:
 - Car locking logic
 - VCR
 - Submission:
 - Pdf format preferred
 - Can be created with any editor, or drawing tool (UML drawing tool, xfig, powerpoint, ...)
 - email as an attachment to luca @ soe

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

4

Assignment - due Oct 10

- Tools track: car locking logic
 - Think at the events that are most relevant:
 - key_in_door, key_out_door
 - key_in_ignition, key_starter, key_run
 - user_lights_on, user_lights_off (input)
 - switch_lights_on, switch_lights_off
 - cabin_light_on, cabin_light_off
 - is_window_open, is_window_closed
 - ...
 - and above all, ring_door_ding !!
 - Design the logic of your dream car

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

5

Assignment - due Oct 10

- Tools track: VCR
 - Think at the various states (taping, programming, ...)
 - Be sure you specify what appears on the display, and what the machine is doing.
 - Can you make one that is easy to program?

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

6

Assignment - due Oct 10

- Project track:
 - Prepare discussion on project requirements and specifications (the report will be due on Oct 15)
 - Discussion will take place:

Social Sciences 2
Rm 141
6-7:45pm (or until done)
October 10

Projects

- What are the projects?
- Who is on each project?
- Let's make a decision now.

Modeling

- Describing a system at a high level of abstraction
 - A model of the system
 - Used for requirements and specification
- Many notations over time
 - State machines
 - Entity-relationship diagrams
 - Dataflow diagrams
 - ... see last lecture ...
 - **UML**

UML

- UML stands for
Unified Modeling Language
- Design by committee
 - Many interest groups participating
 - Everyone wants their favorite approach to be "in"
- Resulting design is huge
 - Many features
 - Many loosely unrelated styles under one roof
- Could also be called
Union of all Modeling Languages

This Lecture

- We discuss
 - Use Case Diagrams
 - Class Diagrams
 - Sequence Diagrams
 - Activity Diagrams
 - State Diagrams
- This is a subset of UML
 - But probably the most used subset

Running Example: Automatic Train

- Consider an unmanned people-mover
 - Aka as in many airports
- Train
 - Moves on a circular track
 - Visits each of two stations in turn
 - Each station has a "request" button
 - To stop at this station
 - Each train has three "request" buttons
 - To stop at a particular station

Use-Cases

- Describe functionality from the user's perspective
- One (or more) use-cases per kind of user
 - May be many kinds in a complex system
- Use-cases capture requirements

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

13

An Example Use-Case in UML

- Name
 - Normal Train Ride
- Actors
 - Passenger
- Entry Condition
 - Passenger at station
- Exit Condition
 - Passenger leaves station

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

14

An Example Use-Case in UML

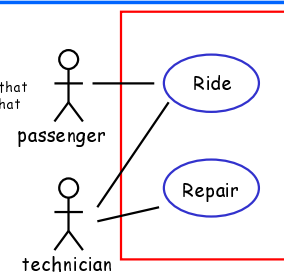
- Event-flow
 - Passenger presses request button
 - Train arrives and stops at platform
 - Doors open
 - Passenger steps into train
 - Doors close
 - Passenger presses request button for final stop
 - ...
 - Doors open at final stop
 - Passenger exits train

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

15

Use Case Diagram

- Graph showing
 - Actors
 - Use cases
 - Edges actor-case if that actor is involved in that case
- Actors
 - Stick figures
- Use cases
 - Ovals



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

16

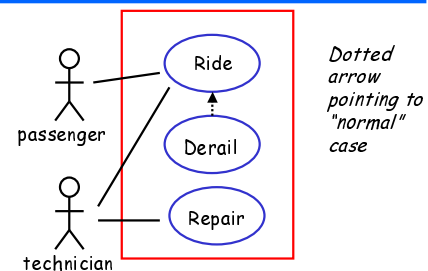
Exceptional Situations

- Some use cases are unusual
 - I.e., error situations
- UML has a special notation
 - The "extends" relationship
 - Nothing to do with OO extension/inheritance
 - These are just rare cases
 - May be nearly unrelated to normal cases

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

17

Extension



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

18

Summary of Use Cases

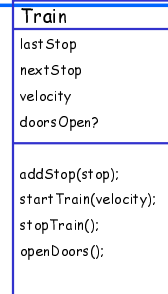
- Use Case Diagram
 - Shows all actors, use cases, relationships
- 5 parts to each use case
 - Name, Actors, Entry/Exit Conditions, Event Flow
 - Actors are agents external to the system
 - E.g., users
 - Event flows are sequence of steps
 - In English

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

19

Class Diagrams

- Describe classes
 - In the OO sense
- Each box is a class
 - List fields
 - List methods
- The more detail, the more like a design it becomes



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

20

Class Diagrams: Relationships

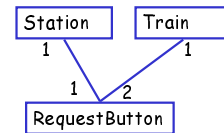
- Many different kinds of edges to show different relationships between classes
- Mention just a couple

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

21

Associations

- Capture n-m relationships
 - Subsumes ER diagrams
- Label endpoints of edge with cardinalities
 - Use * for arbitrary



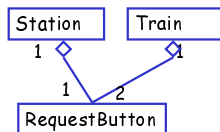
One request button per station; each train has three request buttons

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

22

Aggregation

- Show *contains* a relationships
- Station and Train classes can contain their respective buttons
- Denoted by open diamond on the "contains" side

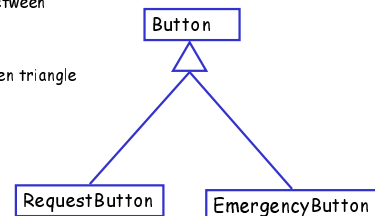


Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

23

Generalization

- Inheritance between classes
- Denoted by open triangle



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

24

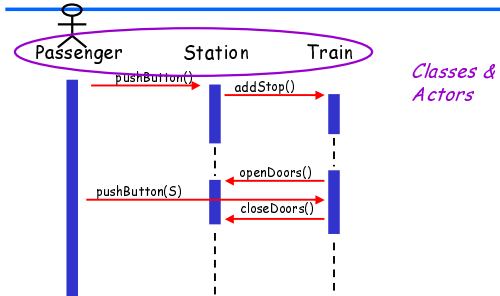
Sequence Diagrams

- A table
 - Columns are classes or actors
 - Rows are time steps
 - Entries show control/data flow
 - Method invocations
 - Important changes in state

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

25

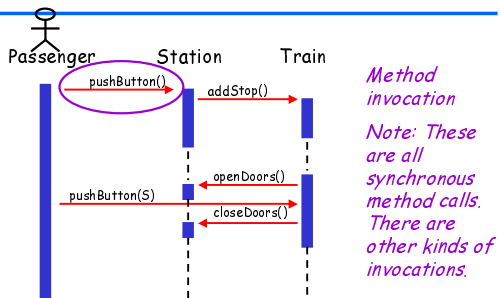
Example Sequence Diagram



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

26

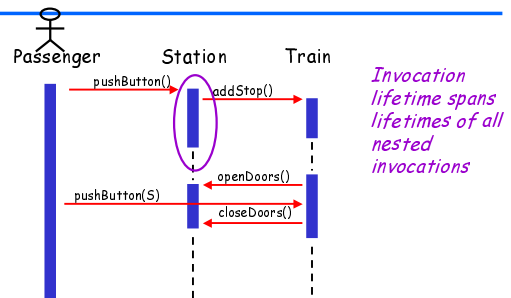
Example Sequence Diagram



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

27

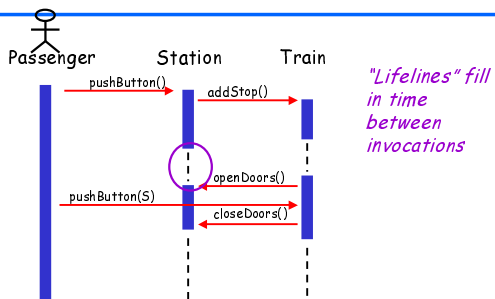
Example Sequence Diagram



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

28

Example Sequence Diagram



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

29

Sequence Diagrams Notes

- Sequence diagrams
 - Refine use cases
 - Gives view of dynamic behavior of classes
 - Class diagrams give the static class structure
- Not orthogonal to other diagrams
 - Overlapping functionality
 - True of all UML diagrams

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

30

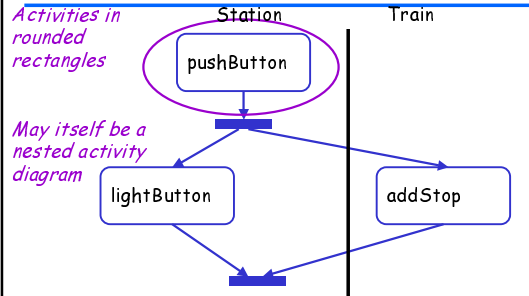
Activity Diagrams

- Reincarnation of flow charts
 - Uses flowchart symbols
- Emphasis on control-flow
- Two useful flowchart extensions
 - Hierarchy
 - A node may be an activity diagram
 - Swim lanes

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

31

Example Activity Diagram

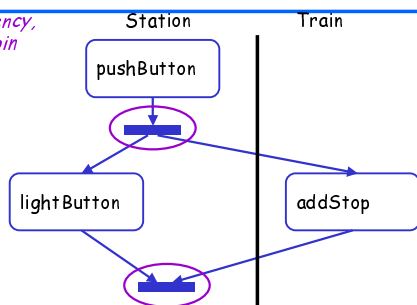


Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

32

Example Activity Diagram

Concurrency, fork & join

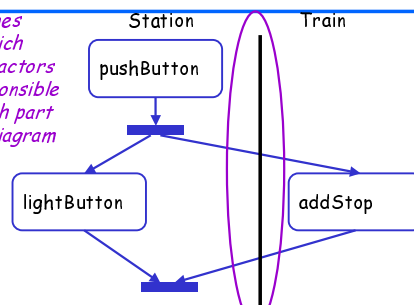


Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

33

Example Activity Diagram

Swim lanes show which classes/actors are responsible for which part of the diagram

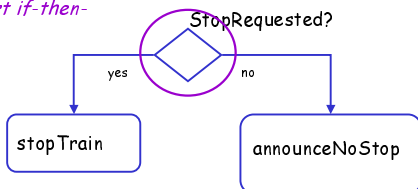


Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

34

Another Example Activity Diagram

Classic flow-chart if-then-else



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

35

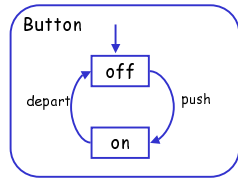
StateCharts

- Hierarchical finite automata
 - Invented by David Harel, 1983
- Specify automata with many states compactly
- Complications in meaning of transitions
 - What it means to enter/exit a compound state

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

36

Example Simple StateChart



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

37

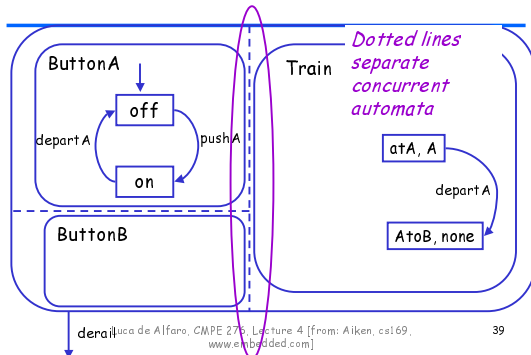
StateChart for the Train

- A train can be
 - At a station
 - Between stations
- Pending requests are subset of {A,B}
- 16 possible states
 - Transitions: pushA, pushB, departA, departB, ...

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

38

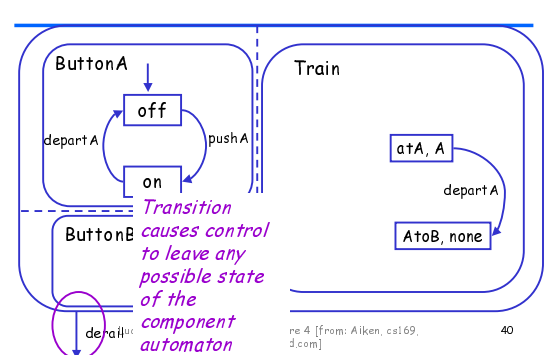
StateChart for Buttons + Train



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

39

StateChart for Buttons + Train



Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

40

Opinions about UML: What's Good

- A common language
 - Makes it easier to share requirements, specs, designs
- Visual syntax is useful, to a point
 - A picture is worth 1000 words
 - For the non-technical, easier to grasp simple diagrams than simple pseudo-code
- To the extent UML is precise, forces clarity
 - Much better than natural language
- Commercial tool support
 - Something natural language could never have

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

41

Opinions On UML: What's Bad

- Hodge-podge of ideas
 - Union of most popular modeling languages
 - Sublanguages remain largely unintegrated
 - Verbose, over-designed.
- Visual syntax does not scale well
 - Many details are hard to depict visually
 - Ad hoc text attached to diagrams
 - No visualization advantage for large diagrams
 - 1000 pictures are very hard to understand
- Semantics is not completely clear
 - Some parts of UML underspecified, inconsistent
 - Plans to fix

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, cs169, www.embedded.com]

42

UML is Happening

- UML is being widely adopted
 - By users
 - By tool vendors
 - By programmers
- A step forward
 - Seems useful
 - First standard for high-levels of software process
 - Expect further evolution, development of UML
- Design interchange
 - XMI (using XML to describe UML designs)

Luca de Alfaro, CMPE 276, Lecture 4 [from: Aiken, csl69, www.embedded.com]

43