

Use Cases and UML

Embedded Interface Design

with **Bruce Montgomery**



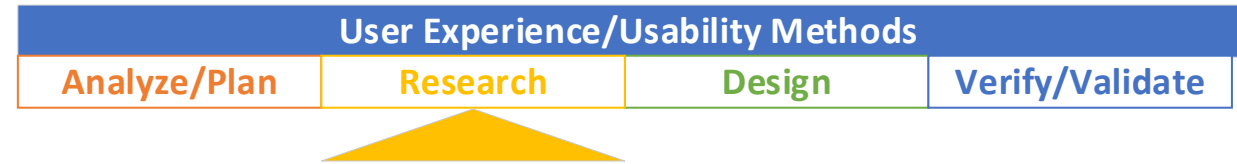
Learning Objectives

Students will be able to...

- Understand how to use and apply text-based use case development and best practices
- Compare and apply the UML-based approach for use cases
- Consider other UML tools for defining system requirements prior to design



Use Cases & UML



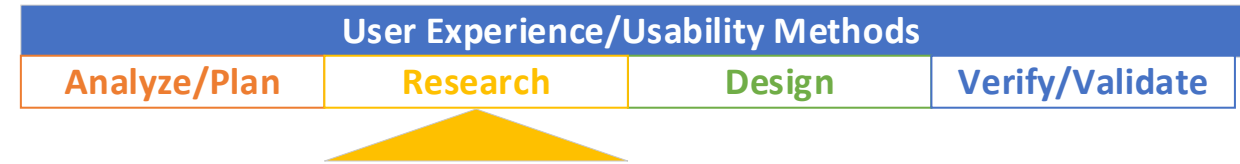
- Use Cases are an important user research tool for helping to understand and document user tasks and scenarios
- Use Cases can be developed using a text-based approach or a specialized UML diagram, we'll look at both
 - UML stands for Unified Modeling Language, we'll look at where it came from
- UML is also useful for capturing other system requirements
 - In particular, we'll introduce UML Sequence, Activity, State, and Class diagrams

Use Cases



- Use cases could be part of both UX Research and Design phases, but more often are part of user and task research
 - Done correctly, use cases bridge business requirements and system design
- You can combine business use cases (broad scope of how actors will use business functions) and system use cases (specifics of functionality from the broader business cases)
- Don't dismiss use cases because they appear easy
- Use case scope is important – use cases must show for a specific scenario, what will the system do, not how
 - Avoid functional decomposition

Text-Based Use Cases

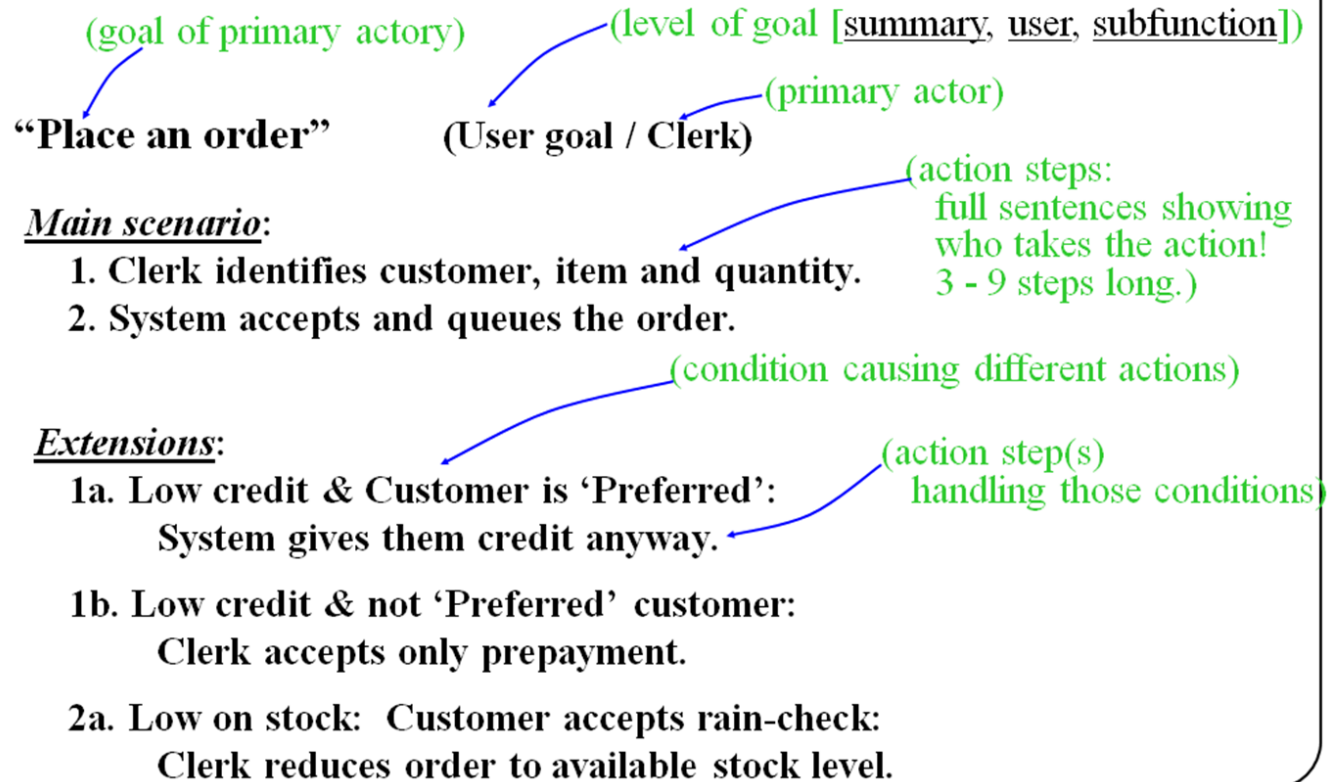


- From a presentation on Agile Use Cases [1] by Alistair Cockburn, author of Writing Effective Use Cases [2]
- What is and isn't a good use case:
 - Good: Text, no GUI, no data formats, 3 to 9 steps in scenario, easy to read, at user's goal level
 - Bad: UML use cases (!), describing a GUI or data, multi-page scenarios, hard/complicated to read
- Use cases can be written up front or as needed
- Use cases can be written completely or incrementally

Example Use Case

Robert Martin: "It shouldn't take longer than 15 minutes to teach someone how to write a use case!"

Use case: Text describing scenarios of user succeeding or failing to achieve goal.



- From Agile Use Cases [1]
- Content: Title, Goal, Level of Goal, Actor(s), Action Steps, Extensions
- Many alternative templates in use

Text-Based Use Cases

- Key elements per [1]
 - Short summary of system goals
 - Main success scenario (system responsibility)
 - Extension conditions (things to watch for or consider)
 - Extension handling (decisions on policy)
- Suggested order of development
 - Start: Use Case, Goal, Scope, Level, Actor, Priority, Frequency
 - Think & Examine: Trigger, Main Success Scenario
 - Check Scope: Extensions, Sub-variations
 - Finish: Performance Target, Open Issues, Schedule, others

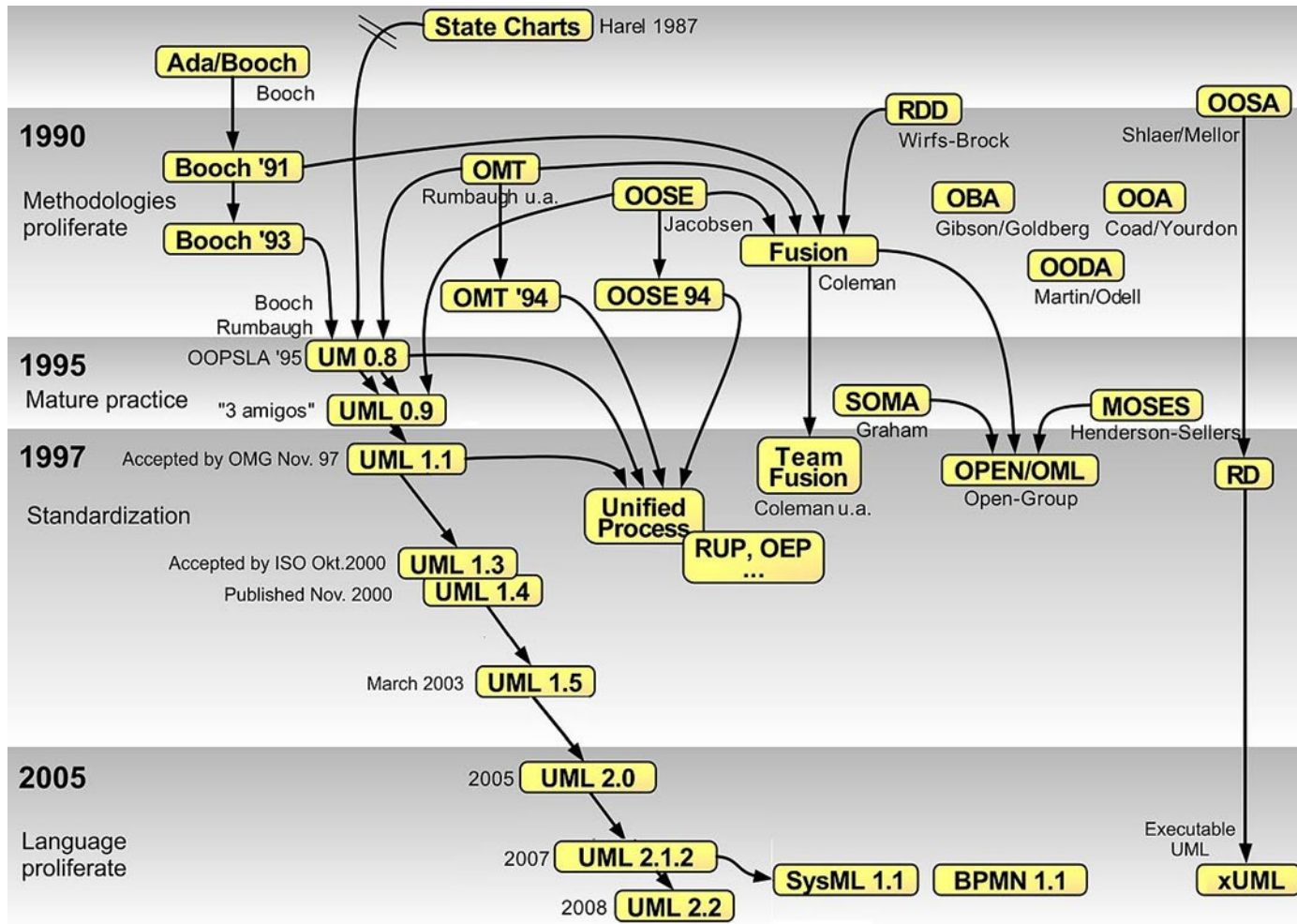


Variety of Use Case Templates

- Cockburn [3] presents a very complex use case template, including sections on characteristic information, main success scenario, extensions, sub-variations, related information, open issues, schedules, etc.
- Typically, most people use a much simpler format, for example:
 - Name, Summary, Actors, Preconditions, Flow, Trigger, Stakeholders
- Here is a reference link to 40(!) different template versions [4]
- My advice is to use the minimum you can to start capturing use cases, then add elements if necessary to your design

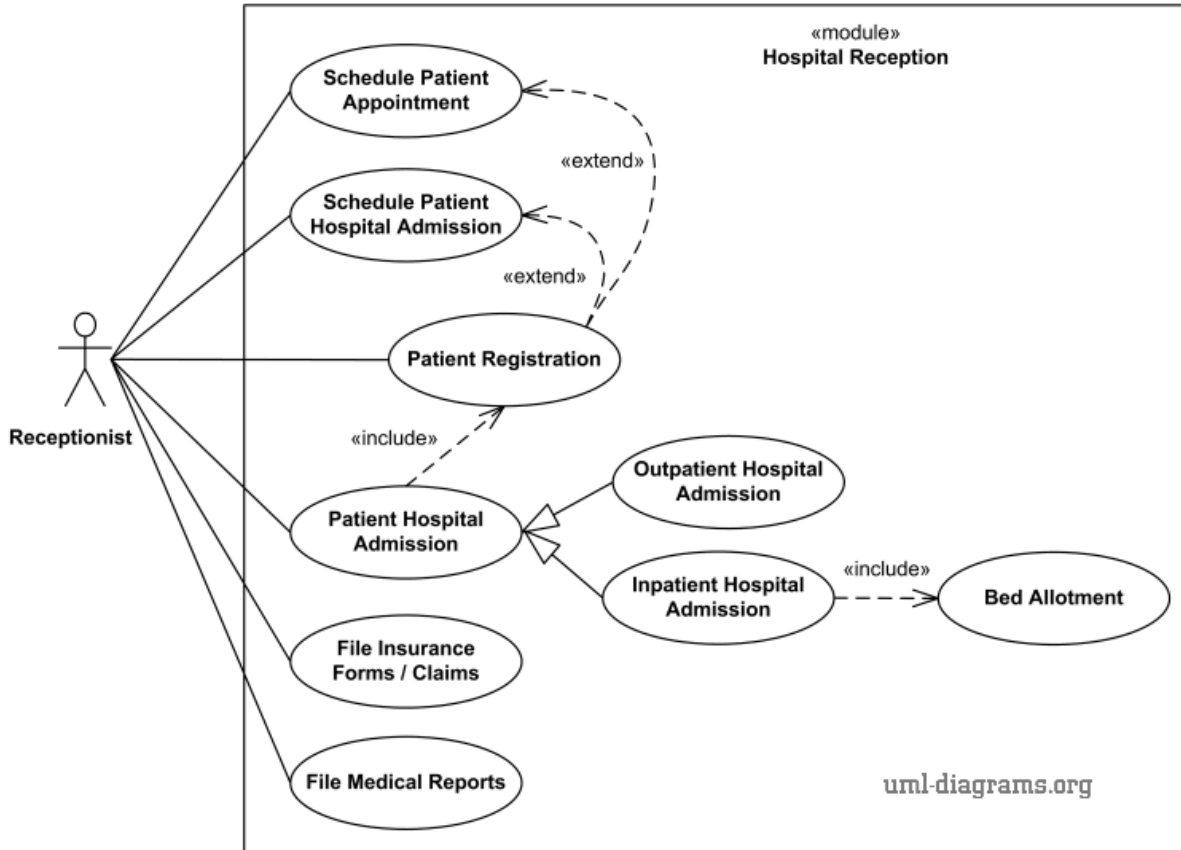


UML - History



- Developed from approaches by Grady Booch, Ivar Jacobson, and James Rumbaugh (the “3 amigos”) at Rational Software
- UML is controlled by the OMG [5]
- Executable and Functional UML – round trip code generation [6]

UML Use Cases



User Experience/Usability Methods

Analyze/Plan

Research

Design

Verify/Validate

- Presents a system from an actor's point of view
- Includes entire flow of a scenario or operation, moving from task to task
- <<Includes>>: non-optional shared or required tasks
- <<Extends>>: optional, conditional tasks
- Example from [7]

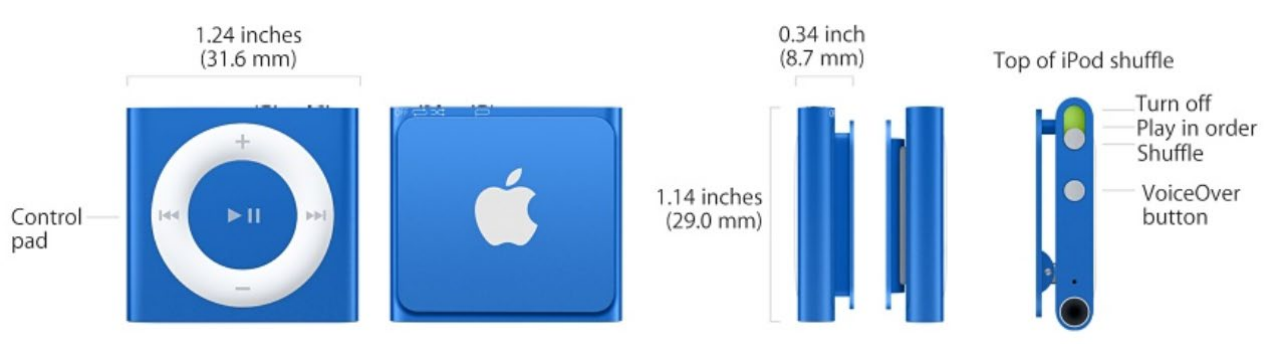
Use Cases Best Practices

- For most systems, use cases should number in the tens, not hundreds
- Remember, design of use cases can iterate as information is found
- WAVE Test for Use Cases (from [8])
 - **W**: Use case describes **WHAT** to do, not how
 - **A**: **ACTOR'S** point of view
 - **V**: Has **VALUE** for actor
 - **E**: Use case models **ENTIRE** scenario



Use Case Exercise

- We'll do this together, text-based first...



WAVE Test for Use Cases (from [7])

W: Use case describes WHAT to do, not how

A: ACTOR'S point of view

V: Has VALUE for actor

E: Use case models ENTIRE scenario

Use Case: <number> <name should be the goal as a short active verb phrase>

CHARACTERISTIC INFORMATION

Goal in Context: <a longer statement of the goal, if needed>

Scope: <what system is being considered black-box under design>

Level: <one of: Summary, Primary task, Sub-function>

Preconditions: <what we expect is already the state of the world>

Success End Condition: <the state of the world upon successful completion>

Failed End Condition: <the state of the world if goal abandoned>

Primary Actor: <a role name for the primary actor, or description>

Trigger: <the action upon the system that starts the use case, may be time event>

MAIN SUCCESS SCENARIO

<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>

<step #> <action description>

EXTENSIONS

<put extensions here, one at a time, each referring to the step of the main scenario>

<step altered> <condition> : <action or sub.use case>

<step altered> <condition> : <action or sub.use case>

SUB-VARIATIONS

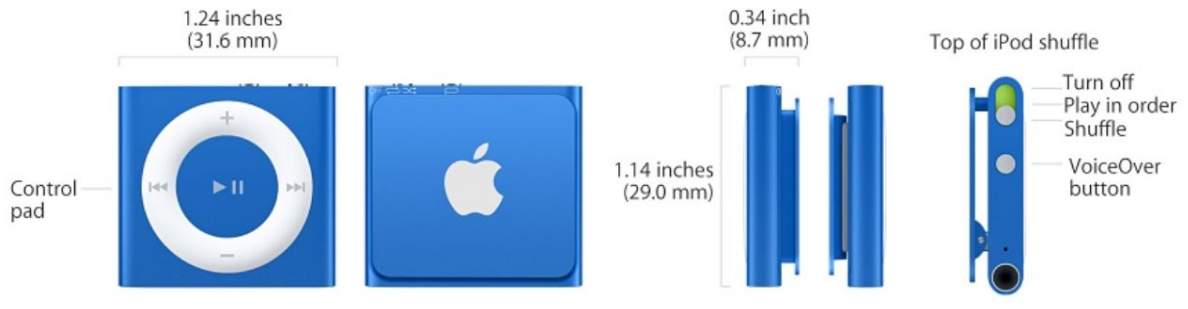
<put here the sub-variations that will cause eventual bifurcation in the scenario>

<step or variation # > <list of sub-variations>

<step or variation # > <list of sub-variations>

Use Case Exercise

- We'll do this together, now UML-based



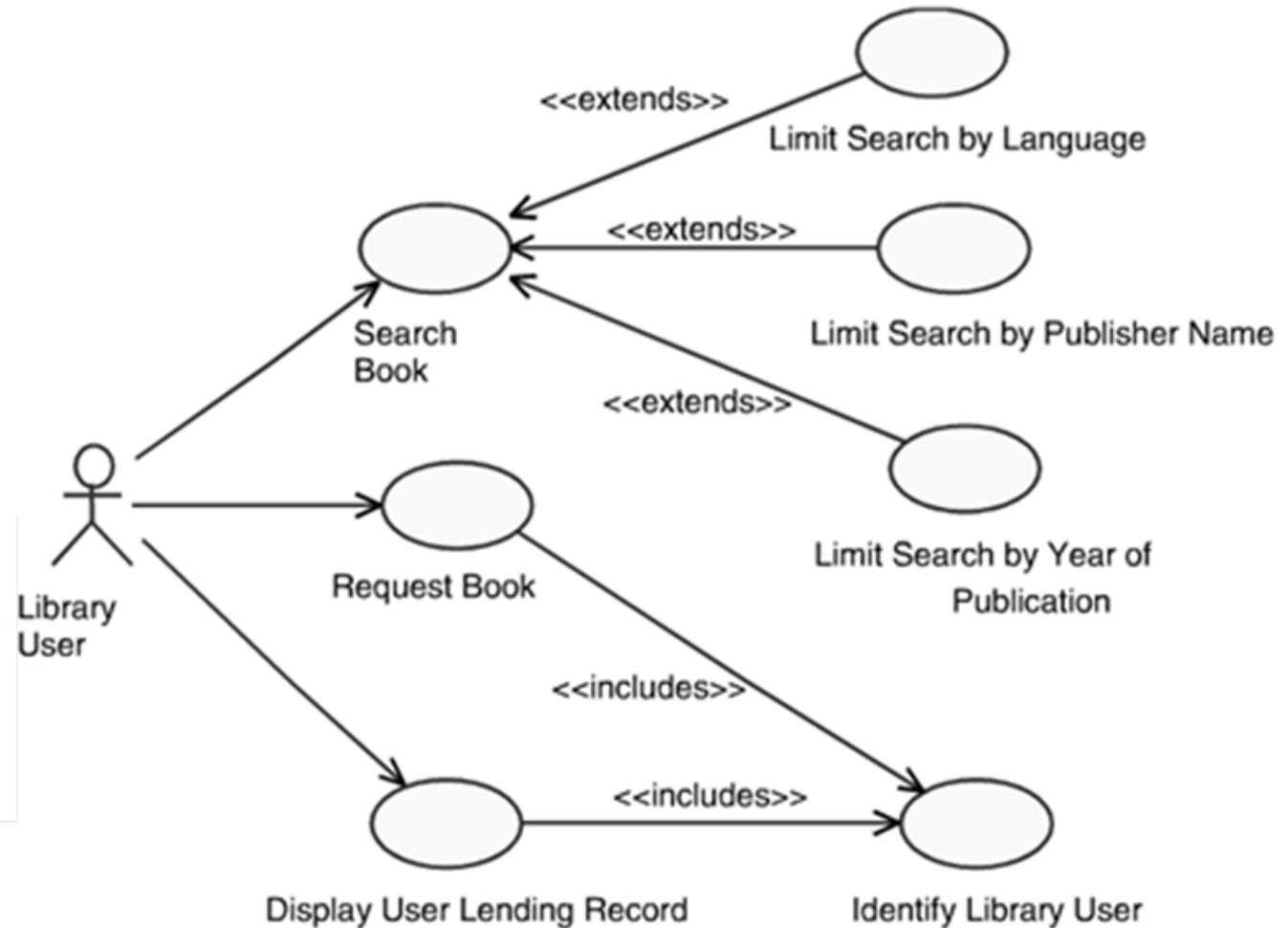
WAVE Test for Use Cases (from [7])

W: Use case describes WHAT to do, not how

A: ACTOR'S point of view

V: Has VALUE for actor

E: Use case models ENTIRE scenario

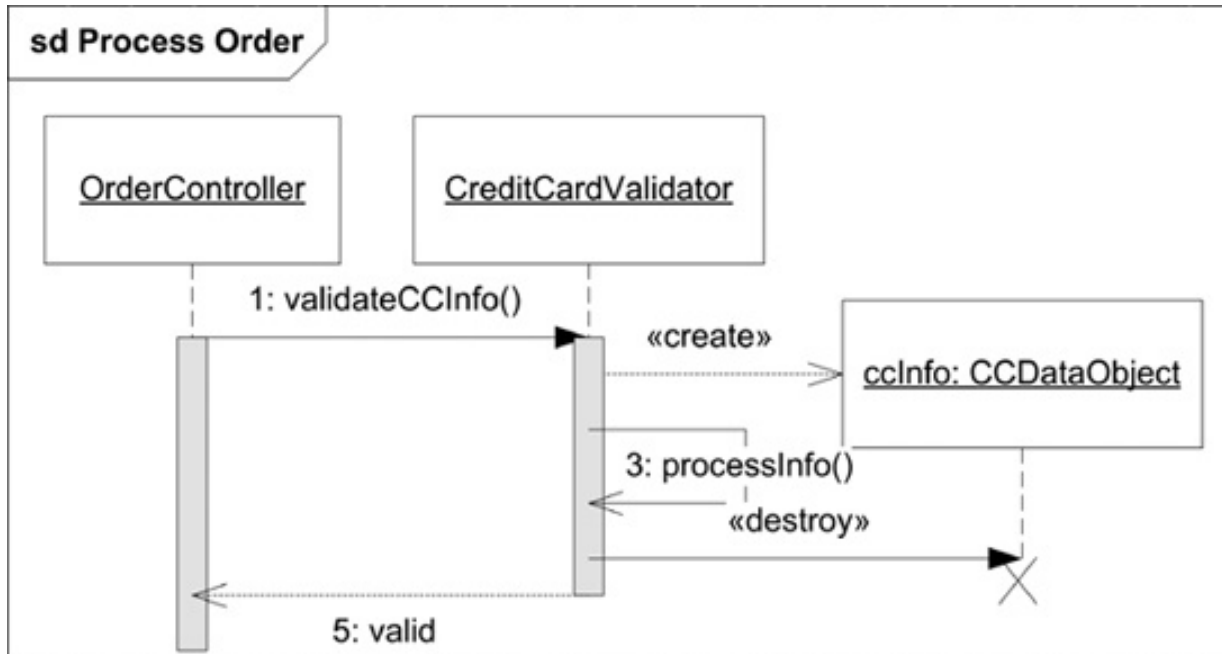


Other Useful UML Diagrams

- The UML 2.5.1 release includes over 20 different types of diagrams [5] in categories including structural, behavioral, and supplemental models
- You may find a use for particular UML diagrams in developing requirements and designs, but there are a number of common diagrams supported by most design tools, including:
 - Use Case
 - Sequence
 - Activity
 - State
 - Class

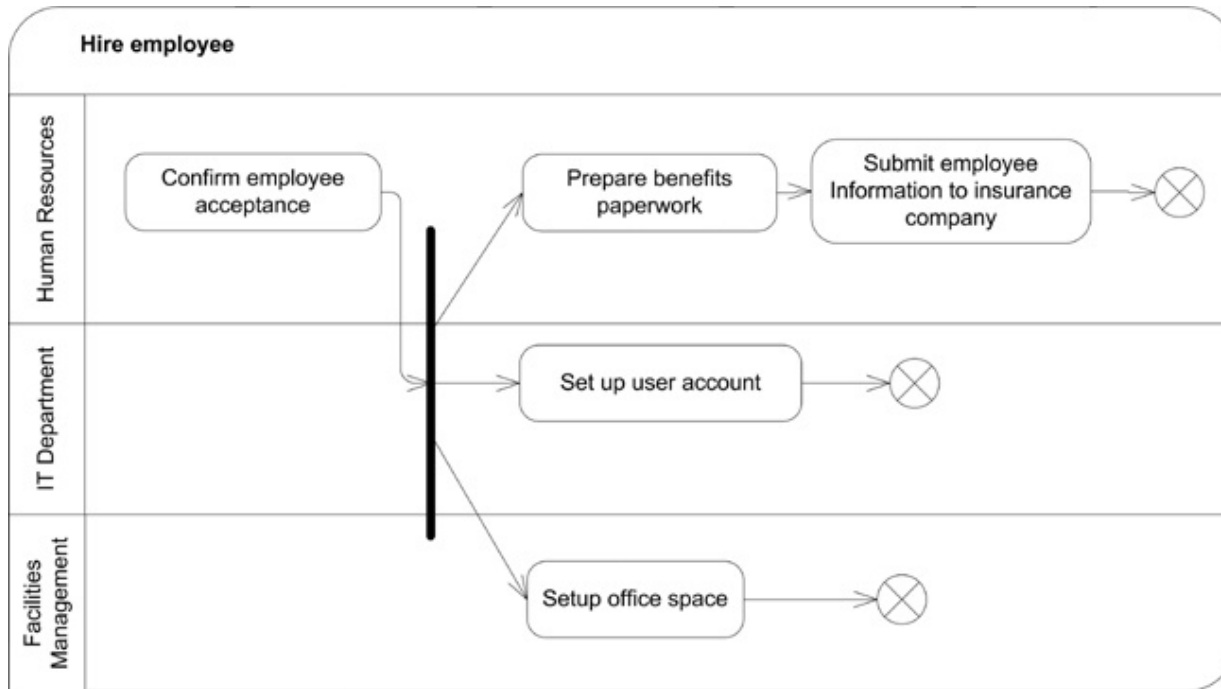


UML Sequence Diagrams



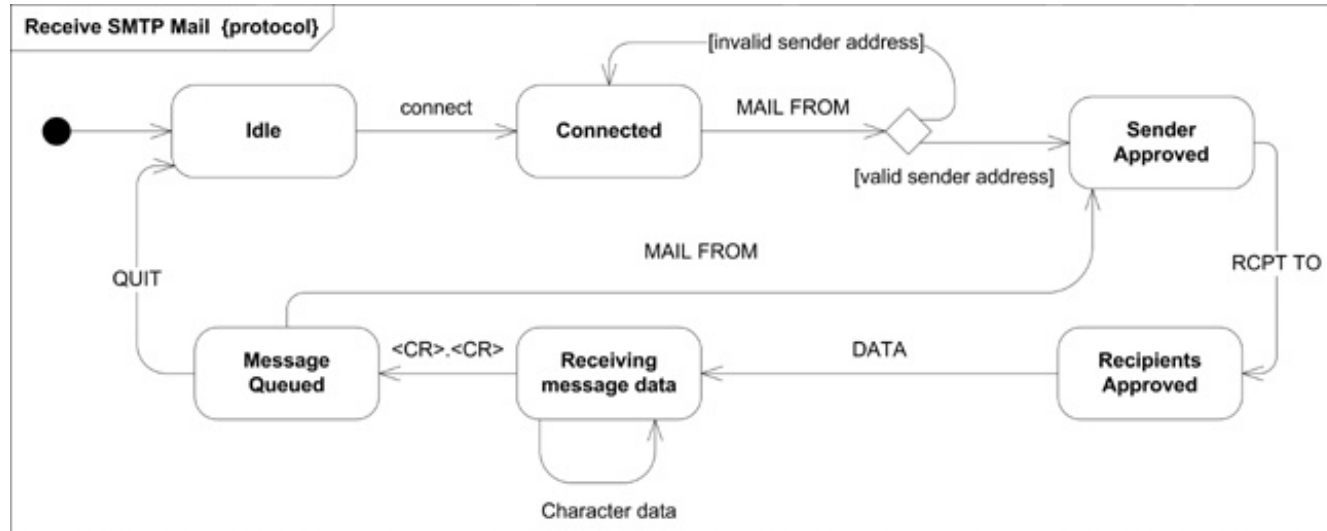
- Sequence Diagrams are generally used to show message transactions between objects in a system over time
- Each object sits atop its lifeline
- By practice, they are drawn with primary flow to the right
- Reference [9]

UML Activity Diagrams



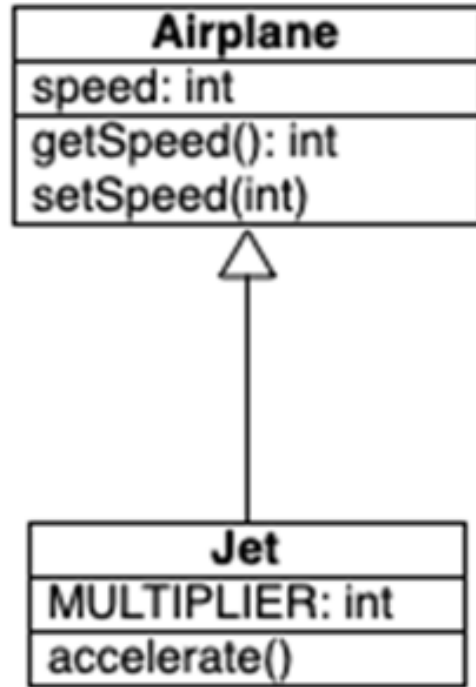
- Activity Diagrams are essentially well structured flow charts
- They can be partitioned to identify separate workflow activities
- Easy to identify start and end points as well as conditional selections
- Reference [9]

UML State Diagrams



- State Diagrams show how systems transition between state nodes
- Often used to develop code-based state machines
- Reference [9]

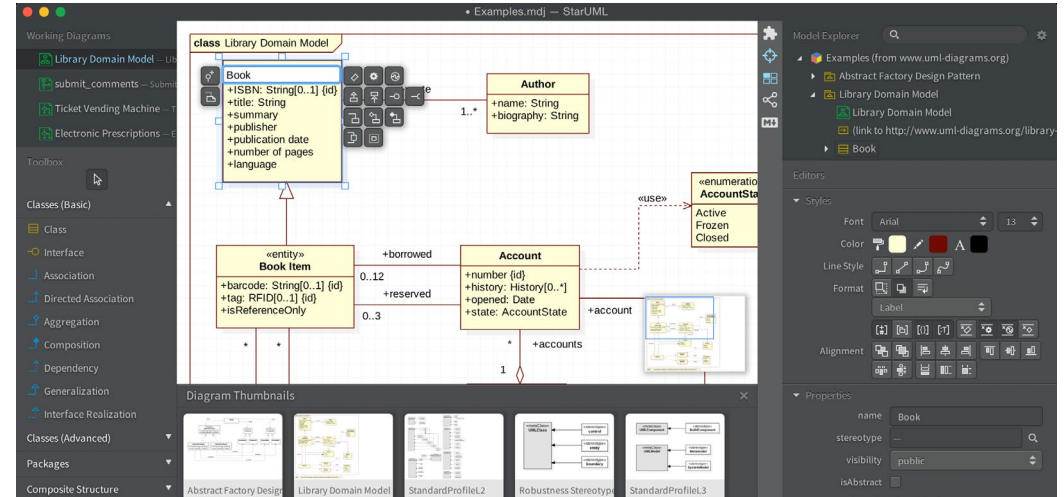
UML Class Diagrams



- Class Diagrams, at varying levels of detail, show the definition of an object-oriented architecture for a system
- Typically used for design in object-oriented languages like C++, Java, C#, etc.
- Diagrams include classes, interfaces, multiplicity, inheritance and aggregation, data attributes and member functions
- If detailed enough, class diagrams can be used to generate code

UML Tools and Resources

- Web-based Tutorials
 - Example tutorial [10]
- Drawing tools
 - Paper & pencil
 - Whiteboard and phone/camera
 - Word processors or drawing tools (e.g. Visio)
 - Specialized UML tools
 - Easily found by web search, typical list [11]
 - Example pictured, StarUML, at [12]
 - Recently, particularly like Draw.IO



Summary

- Use Cases are a very common tool for detailing key user transactions needed to make a successful system
- UML, powered by the picture equaling a thousand words, is often a quicker way to capture use cases and other requirements of a system
- As always, you'll need to consider what makes sense for your system designs in the UX Research and other phases



Midterm Exam

- In class on Wednesday 10/16 (will review topics on Monday 10/14)
- Closed book exam, no connected devices
- You may have one 8.5 x 11 inch note paper, you can use both sides
- You may write out or type your notes
- You cannot use other people's note materials, the notes must be your own
- You will turn in your notes with the exam papers
- You will be given spare paper for scratch paper or for writing out longer answers if needed



Next Steps

- Project 3 due Monday 10/21
- Super Project Proposal (not graded, but required) – Friday 10/11
- Active Quiz due Friday at class time – no new quizzes for next two weeks – leading up to midterm
- EID Midterm is Wednesday 10/16 in class
- Class staff available to help
 - Shubham - Tues 12-2 PM, Fri 3-5 PM in ECEE 1B24
 - Sharanjeet - Tues 2-3 PM, Thur 2-3 PM in ECEE 1B24
 - Bruce - Tue 9:30-10:30 AM, Thur 1-2 PM in ECOT 242
- Final Exam is set
 - Tuesday Dec 17 7:30 PM - 10 PM ECCR 1B51
 - Final will be open notes and Canvas based, you'll need a PC



References

- [1] <http://alistair.cockburn.us/get/2231>
- [2] Writing Effective Use Cases, Cockburn, 2001, Addison-Wesley
- [3] <http://alistair.cockburn.us/Basic+use+case+template>
- [4] <https://templatelab.com/use-case-templates/>
- [5] <http://www.omg.org/>
- [6] <http://modeling-languages.com/list-of-executable-uml-tools/>
- [7] <https://www.uml-diagrams.org/examples/hospital-management-use-case-diagram-example.html>
- [8] UML for Mere Mortals, Maksimchuk & Naiburg, 2005, Addison Wesley
- [9] UML 2.0 Pocket Reference, Pilone, 2006, O'Reilly
- [10] <https://www.tutorialspoint.com/uml/index.htm>
- [11] <https://www.guru99.com/best-uml-tools.html>
- [12] <http://staruml.io/>

