Thursday 9/11/14 ~smb/classes/f14 ≤4187 4156

use cases more detailed functional requirements than user stories
- used in addition or instead
- constructed by developers, not customers
- textual &/or UML

how to write a use case

1. identify who/what will be using the system - actors (not necessarily all stakeholders)

2. pick one

3. identify an objective this user (role) wants to achieve with the system

    - name with verb or verb+noun
    - concrete & specific
    - avoid verbs like "do" & "perform", & nouns like "data" & "information"

4. decide what should be the normal course of events

Scan Notes

Thursday 9/11/14 4156

"happy path" "sunny day"

if there is no single normal course of events, then might be unintentionally combining multiple use cases

may have preconditions

5. describe what the user does as a sequence of steps, & what the system does in response to each step that user needs to be aware of

usually time ordered

avoid if-then

★ must provide result of value to the user

6. extend with alternate courses of events - include special cases & what can go wrong
<< extend >>

different actor choices

when & end of starts wrt main flow

7. look for commonalities among already defined use cases, & combine into separate use cases << include >>

8. repeat 2-7

example in courseworks
househeeper does laundry

simple, medium, heavyweight versions

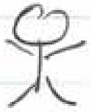
Thursday 9/11/14 4156

Use case textual descriptions often accompanied by UML use case diagrams

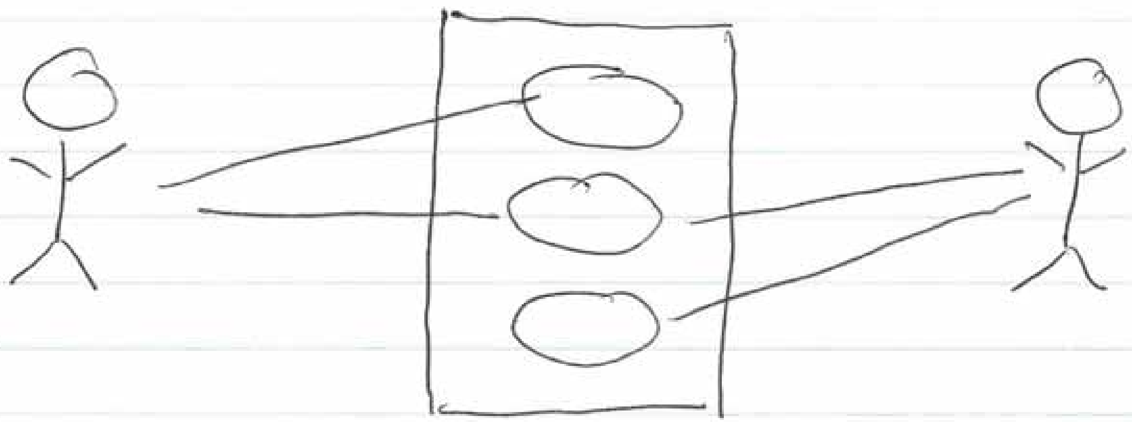
UML = Unified Modeling Language
unified concepts from various OO modeling approaches in mid 1990s

helps communicate requirements at high level & depict scope
- actors outside system
- use cases inside

Use case oval

actor stick figure 

system boundary rectangle

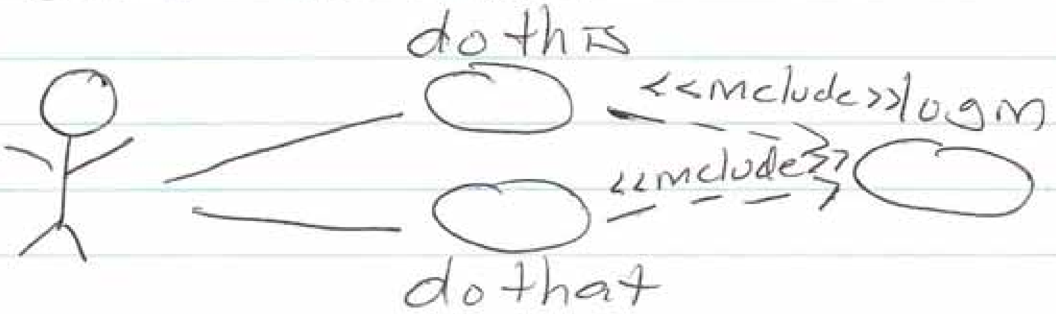


Thursday 9/11/14 4156

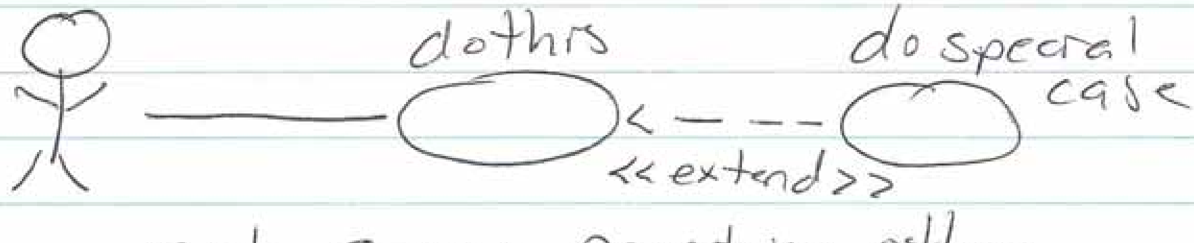
lines depict relationships between actors & use cases, & between different use cases

default relationship is communication request information from or invoke the use case

include relationship
- use case included in two or more other use cases



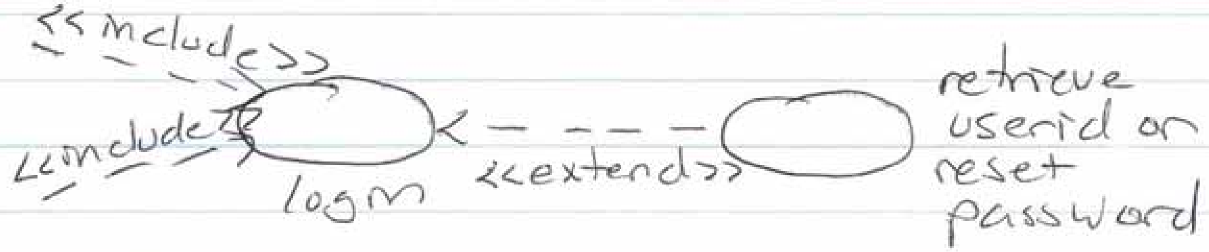
extend relationship
exceptional variation or alternative flow
(may have own alternative subflows)



note arrow points in other direction

Thursday 9/11/14 4156

example - User has forgotten
Userid &/or password



tip: avoid CRUD
create, read, update, delete
focus on what actor really wants to do

example - university registration system
involves creating, modifying, & deleting schedules
but student actor does not care about schedules in the database, only wants to register for courses

tip: do not reference UI elements like pages or buttons

tip: avoid architectural details
No schedule is saved in the MySQL database