Overview of Use Cases

- Many different flavours of use cases
- many discussions, articles, books but few or differing definitions of what a use case is
  - part of its success?
- Jacobson's original was more restrictive, formal and user centric … has become less formal, open to interpretation, system oriented ?

Example of system overview
NOT *a use case*
(Constantine and Lockwood)

"The guest makes a reservation with the hotel. The hotel will take as many reservations as it has rooms available. When a guest arrives, he or she is processed by the registration clerk. The clerk will check the details provided by the guest with those that are already recorded. Sometimes guests do not make a reservation before they arrive. Some guests want to stay in non-smoking rooms.

What is the use case here, and what is its purpose? Is it to reserve a room or to obtain a room?Or is it to check in a guest? Who is the user and in what is role do they act? Is the user a guest or the clerk or the telephone operator who takes the reservation? What is the interface to be designed?"

Use Case definitions

- [Rumbaugh et al] The specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside actors … that yields an observable result of value to a particular actor
- [Martin Fowler]: A use case is a typical interaction between a user and a computer system … [that] captures some user-visible function … [and] achieves a discrete goal for the user.

Use Cases: Our approach

Following Alistair Cockburn and Ivor Jacobson guidelines, and similar to essential use cases of Constantine and Lockwood

- Main purposes
  - to capture the system requirements
  - provide starting point for design
  - act as a reference to compare design to
  - basis for test sets
  - ….

Use Cases: Our approach

Following Larman, Cockburn and Jacobson and others,

Use case text focuses on:

- the interactions between actor and system
- in order to achieve a goal of the actor
- described in a technology and implementation independent manner

Use Cases

actor

- external entity that interacts with the system
  - e.g. human, software, physical
- role rather than an individual e.g. one human can be a user actor and manager actor

use case

- a set of interactions between the system and one or more actors to achieve some specific goal
- focus on user's goals

- Jacobson *"a sequence of transactions performed by a system, which yields an observable result of value for a particular actor"*
- Use case: *"A case of use of the system"*

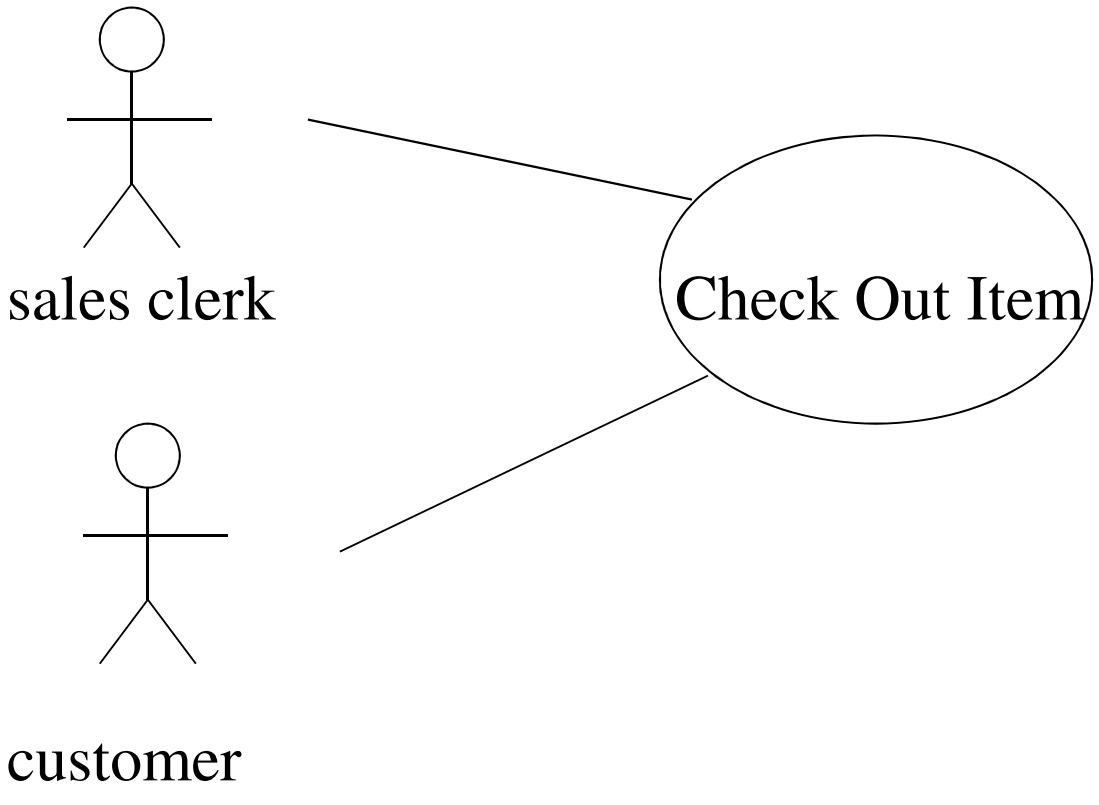
Use cases

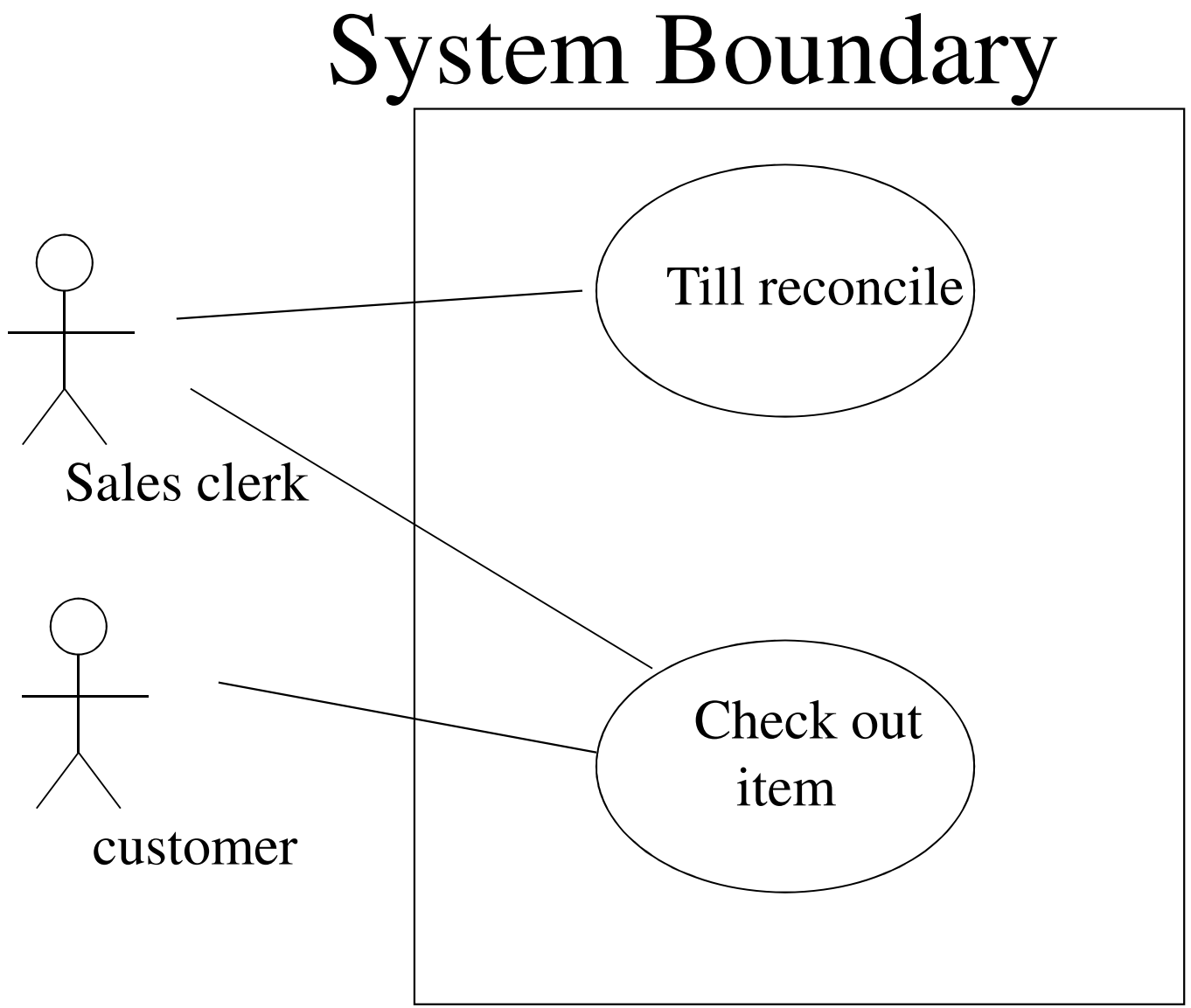
Use case diagram: overview of all use cases and their relationships

Use case text: textual descriptions of individual use cases (the important part)

- **Short textual format:**
  - name the use case;
  - provide an overview, stating the goal;
  - list the actors;
  - state the sequence of interactions in natural language.

UML Use Case Diagrams




System Boundary
Till reconcile
Sales clerk
Check out item
customer

Scenarios

- A use case describes a set of sequences where each sequence represents one scenario
- Scenario is one specific sequence of actions
- scenario is an instance of a use case

A single use case, Search library catalogue,would have many scenarios: searching by keyword, searching by author, searching by title ...

Relationships in use cases

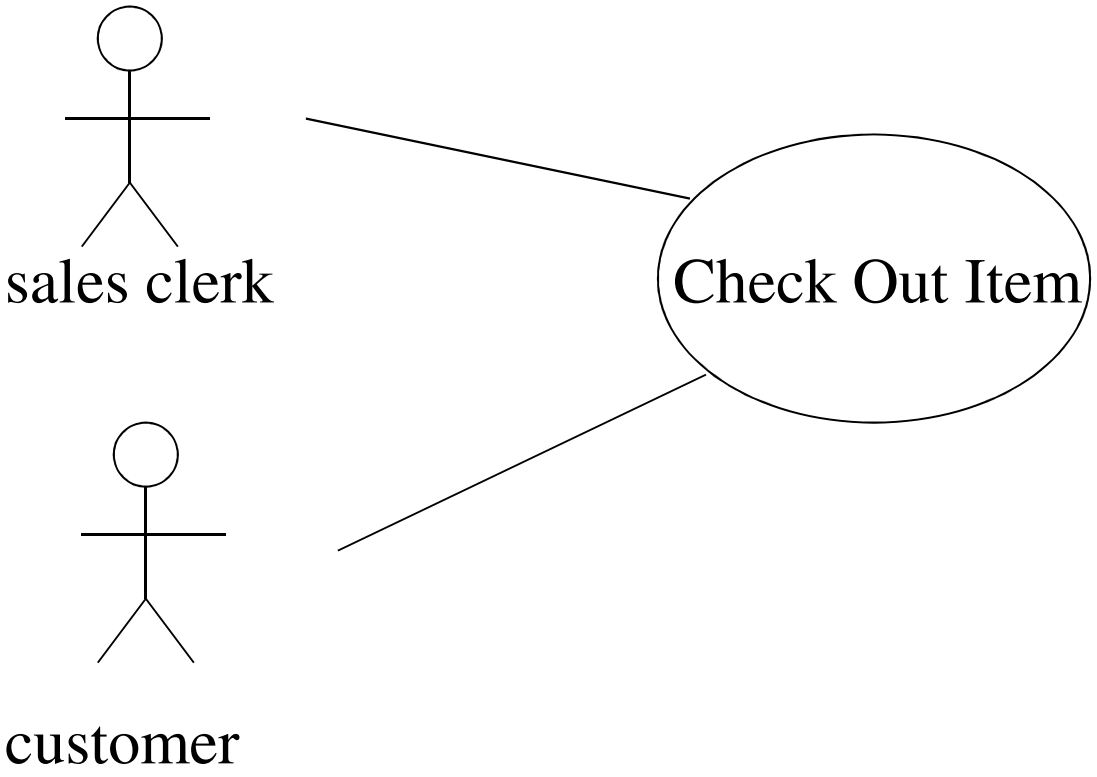
- Association
  - actors associated with use cases
- generalisation
  - inheritance relationship, e.g. between use cases, between actors

Relationships in use case

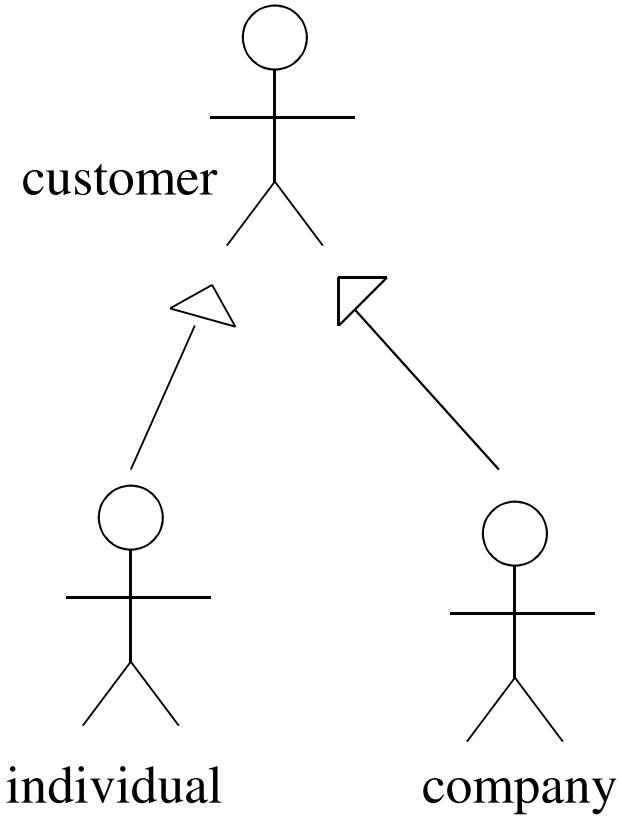
- <<extend>>
  - a use case is extended to describe more behaviour
  - extra behaviour required by actor
- <<include>>
  - a use case includes another use case
  - common behaviour in system

Association

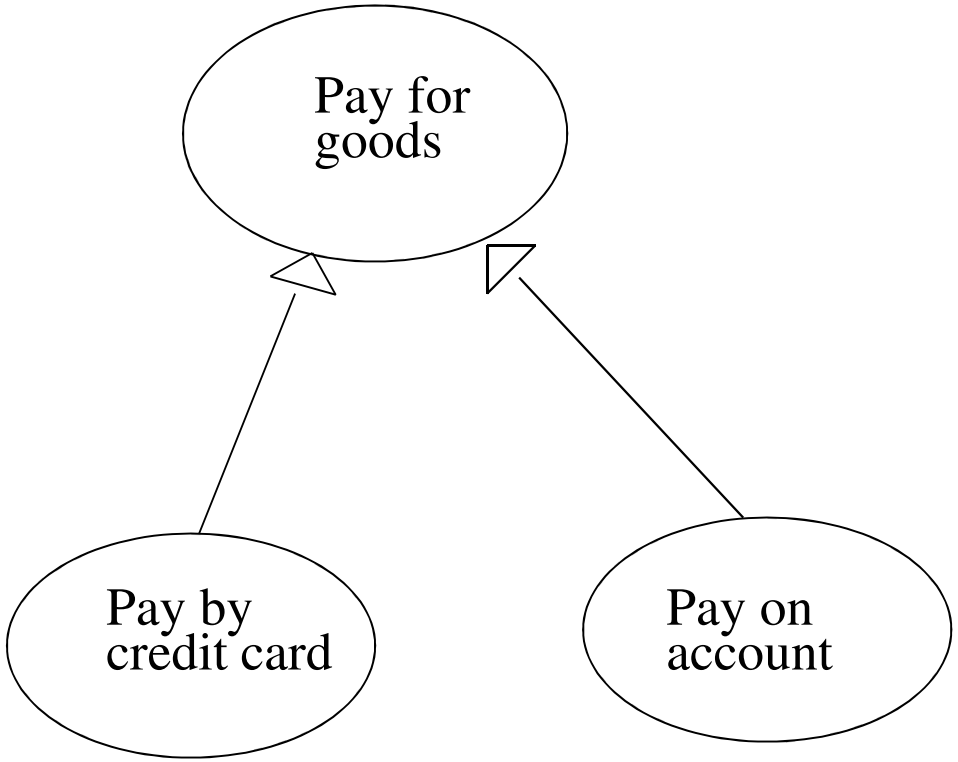
denoted by line connecting actors to use case



Generalisation



Actor generalisation

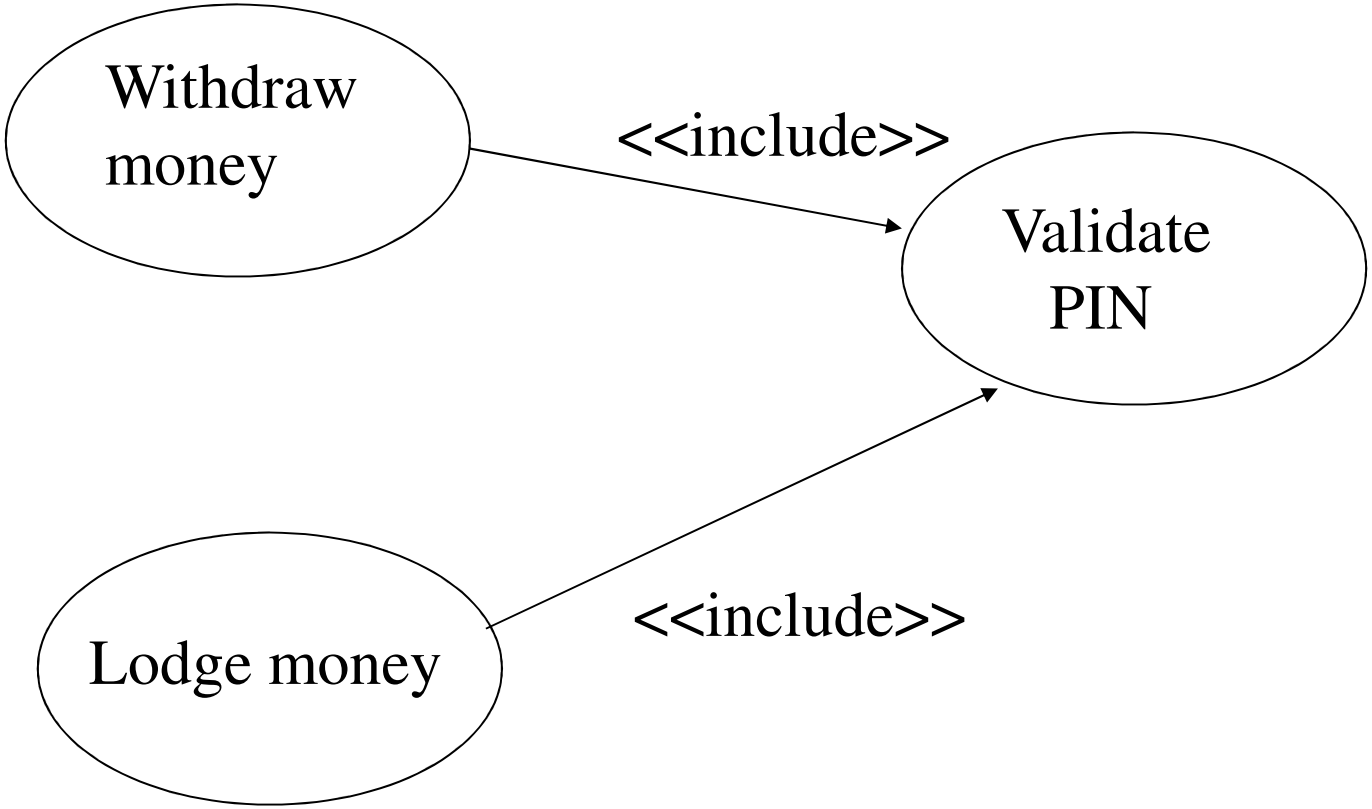


Use case generalisation

include

- Identifies common parts
- makes use cases more compact
- prepares for analysis

include



include

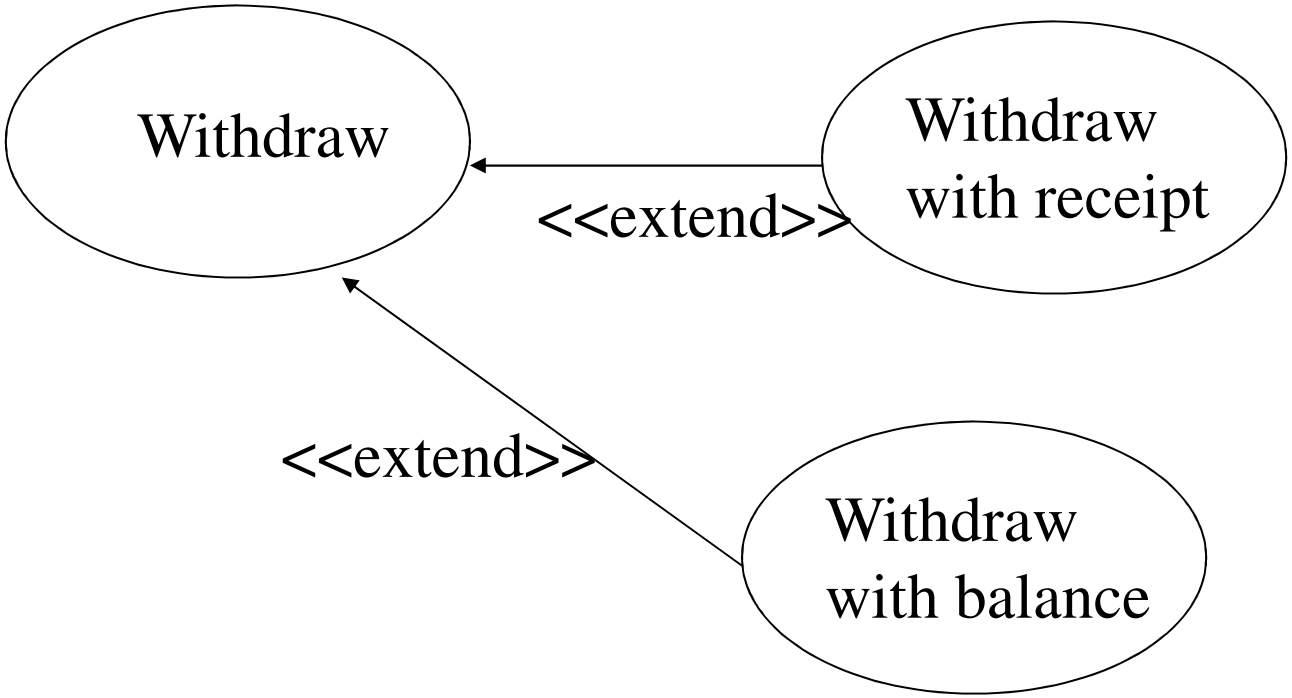
- Document textually
  - use *name of sub use case*
  - describe the sub use case separately
- Always useful to rationalise and simplify using common sub use cases

Extend

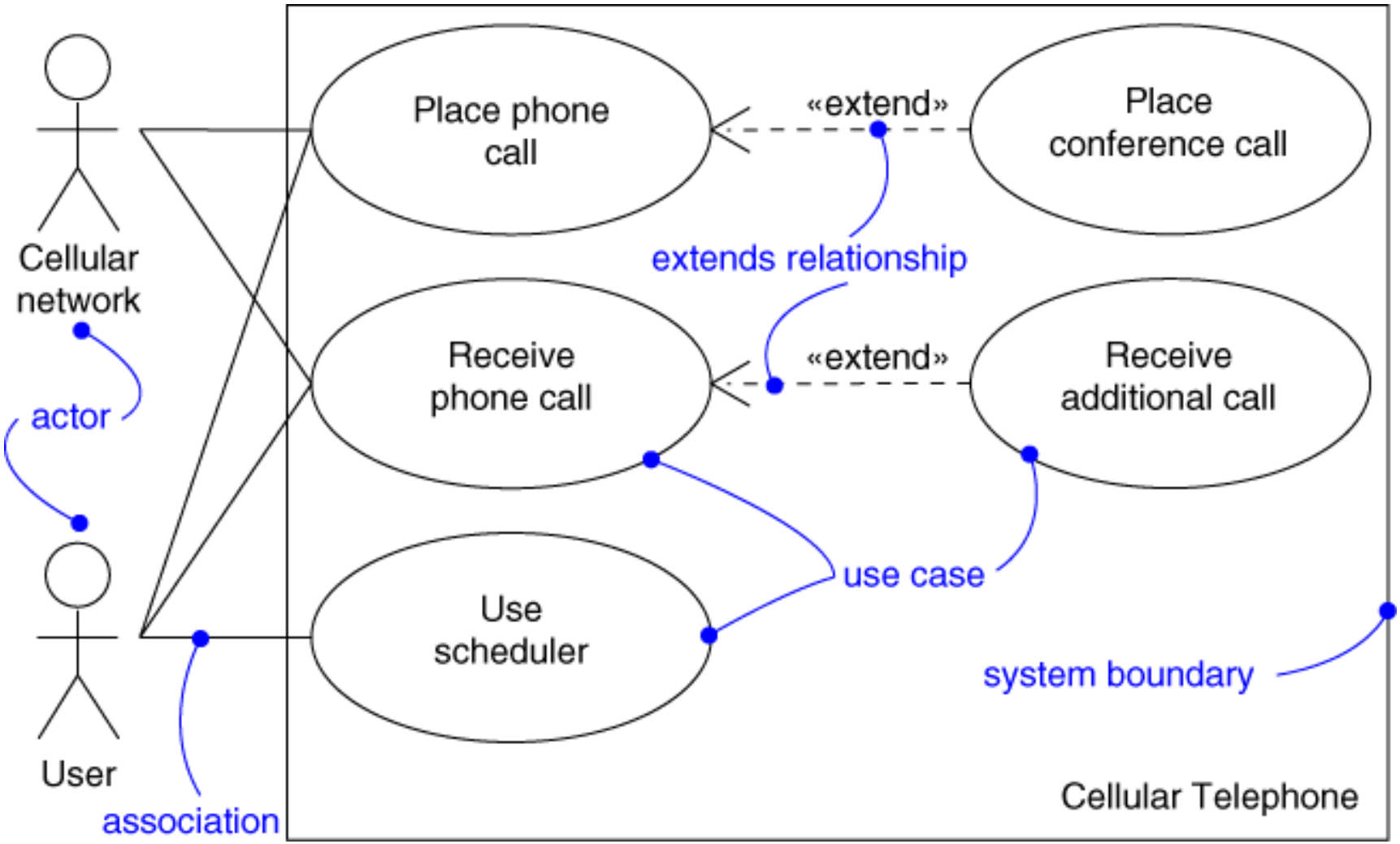
- Models optional parts of use case
- models rare, complex variations of use case
- models error or exception handling
- introduces modularity into use case

•Document textually
  - label the point of departure
  - describe extension
•Could avoid extend by including in the main use case using conditional *(if … then …)*
•… no right way to do use cases ...

extend


Withdraw
Withdraw
with receipt
<<extend>>
<<extend>>
Withdraw
with balance

Use Case Diagram



Use Cases:
Examples and Guidelines

Use Cases

- Capture requirements
- Provide a basis for (and used in) analysis
- Act as reference for: Architecture; Design; Services/implementation; Testing

Amount of detail required:

  – Examples 1,2,3 indicate suitable level of detail

Capturing System Requirements

When looking at various software development processes will see the role of use cases in specifying the system requirements.

Example method:

- Decompose each high-level functional requirement of into a set of use cases covering the main interactions with the system

"A use case captures the intended behavior of the system (or subsystem, class, or interface) you are developing, without having to specify **how** the behavior is implemented. That's an important separation because the analysis of a system (which specifies behavior) should, as much as possible, not be influenced by implementation issues (which specify how that behavior is to be carried out)." From UML User guide [Booch, Rumbaugh, Jacobson]

Use Case: example 1

- Use case name: Check Out Item
- Actors: customer; sales clerk
- Interactions:
  - customer presents item
  - sales clerk swipes barcode reader across item
  - system looks up barcode in database
  - system displays item description and price
  - system adds item and price to invoice
  - system displays current running total

Use Case: example 2

- Use case name: Perform ATM transaction
- Actors: customer
- Interactions:
  - customer inputs card
  - customer enters PIN
  - system validates PIN
  - customer selects transaction
  - system performs transaction
  - system returns card
  - customer takes card

Use Case: example 3

- Use case name: Do on-line purchase
- Actors: customer; warehouse; shipper; credit card company
- Interactions:
  - customer requests item, gives card details
  - system checks item in stock
  - system gets authorization from CC company
  - system requests item from warehouse
  - system requests shipper to deliver item
  - customer receives item, signs for delivery
  - CC company debits customer account

- Use Case: **Buy Items with Cash** (essential use case)
- Actors: Customer (Initiator)
- Purpose: Capture a sale and its cash payment.

Overview: A Customer arrives at a checkout with items to purchase. The cashier records the purchase items and collects a cash payment. On completion, the Customer leaves with the items and a receipt.

- Type: Primary and essential.
- Functions: R1.1, R1.2, R1.3, R1.7, R1.9, R2.1 (Cross References)

- **Typical Course of Events:**

Actor Action	**System Response**
1. This use case begins when a Customer arrives at a POST checkout with items to purchase.	
2. The Customer gives UPC to the system. If there is more than one item, the Customer can give the quantity as well.	3. Displays the price and running total.
4. The customer states there are no more items.	5. Displays the sale total.
6. The customer makes a payment.	7. Displays the balance due and prints a receipt.
8. The Customer leaves with the items purchased, change, and receipt.	

How many?

- How many interactions in a sequence for one use case?
  - Manageable, informative, e.g. 3-11
- How many use cases for a project ?
  - Manageable, informative
  - hierarchy for non-trivial systems
  - project level suggestions range from 20 to several hundred

Discovering Use Cases

- Mission goal decomposition
  - hierarchical structured analysis
- Scenario discovery
  - enumerate system activity scenarios
- Actor centered discovery
  - define actors;
  - find their roles, what tasks they require
  - define responsibilities and collaborators

Use case development

- Iterative process
- Cover all behaviour, emphasising most important aspects
- Cover all?
  - Check all modalities or combinations as to how various parameters may vary … e.g. covering all possible actor interactions in start-up, normal running, maintenance, etc … modes

Alistair Cockburn's reminders

- **Reminder 1**.

  A use case is a prose essay

  – prose more important than diagram

- **Reminder 2.**

  Make the use case easy to read.

  - Keep matters short and too the point.
  - Start from the top and create a coherent story line. The top will be a strategic use case.
  - Name the use cases with short verb phrases that announce the goal to be achieved.

Make the use case easy to read ...

- Start from the trigger, continue until the goal is delivered or abandoned, and the system has done
- Write full sentences with active verb phrases that describe the subgoals getting completed.
- Make sure the actor is visible in each step.
- Make the failure conditions stand out, and their recovery actions readable. Let it be clear what happens next, preferably without having to name step numbers.
- Put alternative behaviors in the extensions, rather than in *if* statements in the main body.
- Create extension use cases only under very selected circumstances.

Alistair Cockburn's reminders

- **Reminder 3. Just one sentence form**
  - a sentence in the present tense,
  - with an active verb in the active voice, describing an actor successfully achieving a goal that moves the process forward.
- Examples
  - "Customer enters card and PIN."
  - "System validates that customer is authorized and has sufficient funds."
  - "PAF system intercepts responses from the web site, and updates the user's portfolio."

Alistair Cockburn's reminders

Reminder 4. Include sub use cases

- macro-like sub use cases
- As a first rule of thumb, always use the *includes* relation between use cases.
- … rather than look for extends and specialises

- **Reminder 5. Who has the ball?**
- Not passive "Credit limit gets entered.",
- make clear what actor initiates
- make clear responsibility

Alistair Cockburn's reminders

Reminder 6. Get the goal level right

- Recall that most use cases have 3-9 steps in the main success scenario, and that the goal level of a step is typically just below the goal level of the use case.

Reminder 7. Keep the GUI out

Reminder 8. Two endings

Reminder 9. Stakeholders need guarantees

Reminder 10. Preconditions

Reminder 11. Pass-fail tests for one use case

Reminder 17.
Work breadth first

- **1 Primary actor**s. Collect all the primary actors as the first step … brainstorm these actors to help you get the most goals on the first round.
- **2 Goal**s. Listing all the goals of all the primary actors
- **3 Main success scenari**o. The main success scenario is typically short and fairly obvious.
- **4 Failure / Extension condition**s. Capture all the extension conditions before worrying about how to handle them.

Agile alternative to use cases

- Many alternatives for documenting requirements

- Scrum and Extreme Programming(XP) have the simpler concept of **user stories**

User Stories in Agile Process

- document customer requirements in XP
- short - one or two sentences
- in the normal language of the user
  - easy for customer to add/change requirements
- user story broken into tasks
- basis for estimating effort, planning

Various sources of info including Scott Ambler
http://www.agilemodeling.com/artifacts/userStory.htm

Examples of User Stories

(www.agilemodeling.com)

- Students can purchase monthly parking passes online.
- Parking passes can be paid via credit cards.
- Parking passes can be paid via PayPal.
- Professors can input student marks.
- Students can obtain their current seminar schedule.
- Students can order official transcripts.
- Students can only enroll in seminars for which they have prerequisites.
- Transcripts will be available online via a standard browser

Summary

Larman use cases

- Very different from user stories
- Have a different purpose
- Have a much bigger role in design
- Have much more defined content and stylr
- ….