CS4125
SYSTEMS ANALYSIS
SPRING SEMESTER 2010-2011

J.J. Collins
Dept of CSIS
University of Limerick

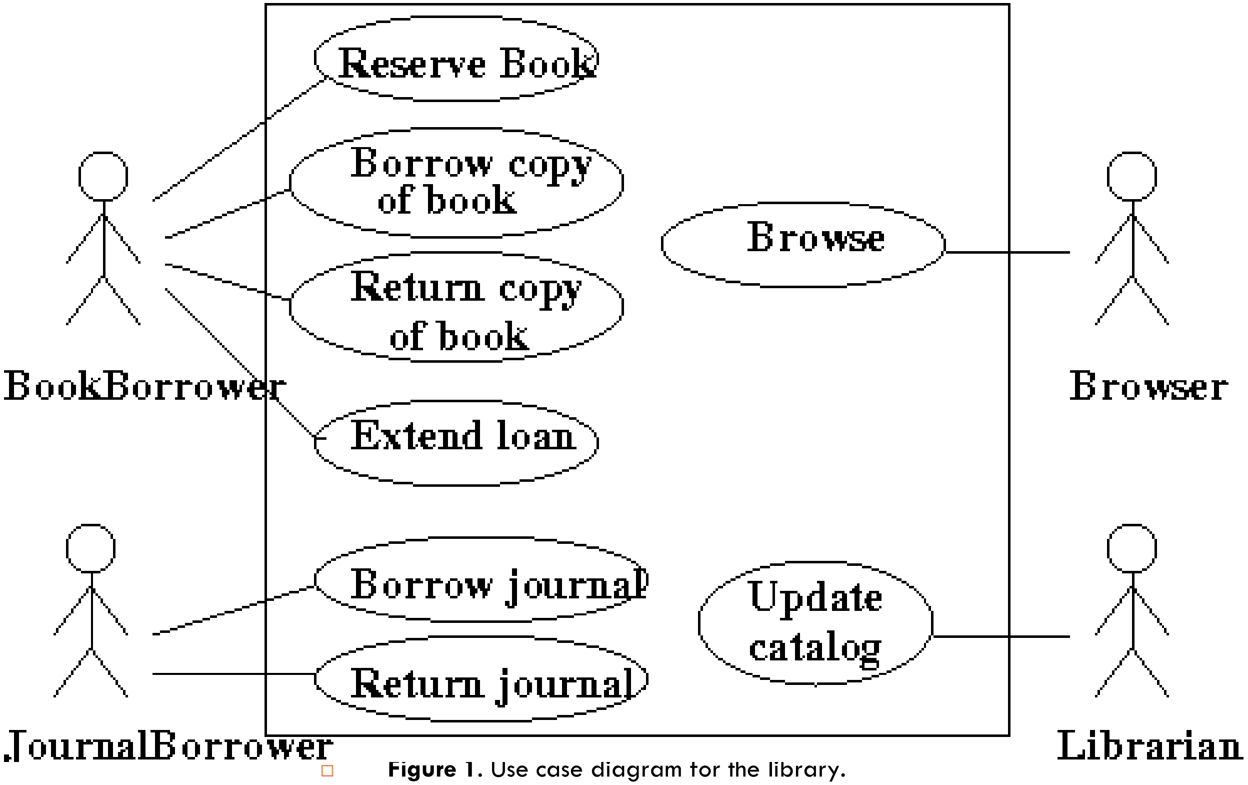
Lecture F (W4L1)

Overview on Use Case Engineering

1. Recap

Lecture	
A	•Craftsmanship → Mature Engineering Discipline using MODELS •Module Themes: •OOAD using the UML, etc. •Architectural and design patterns •Leading to Dependency management and resuse
B	Characteristics of Good Quality Software Contracts and Interfaces Architecture- Centric Component-Based Deveopment (CBD)
C	OO Paradigm Part 1
D	OO Paradigm Part 2
Project Spec	
E	Requirements Engineering Part 1

1. Recap - UML Modelling Example



□ **Figure 1.** Use case diagram for the library.

Borrow copy of book: A BookBorrower presents a book. The system checks that the potential borrower is a member of the library...........

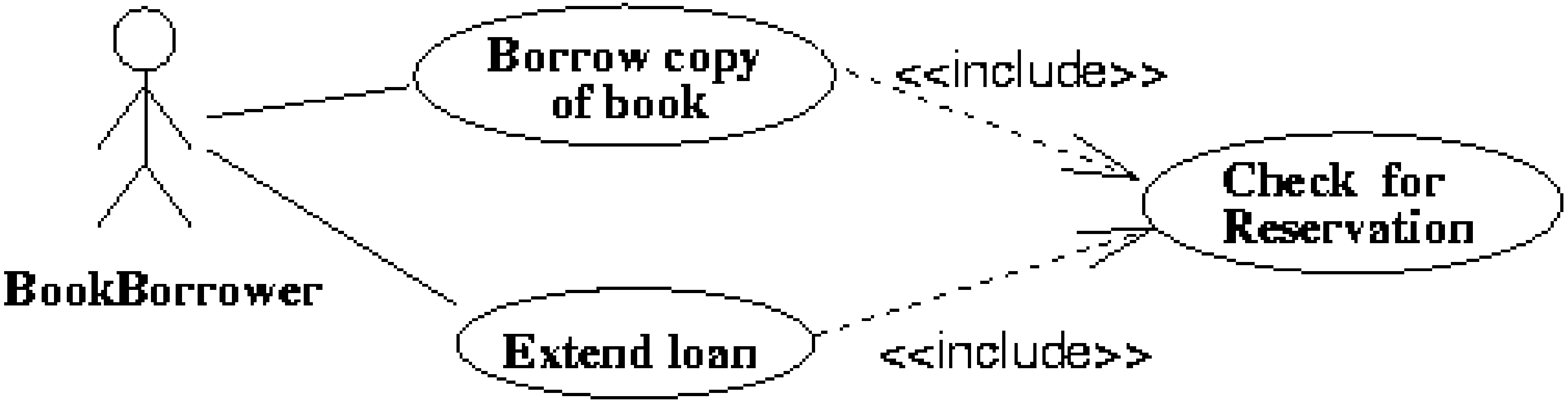
2. Scenarios

- A scenario is an instance of a use case.
- A scenario is a possible interaction between the system and users in their various roles.
- Example: use case “Borrow copy of book”
  - Book borrower Mary borrows the library's third copy of UML distilled, when she has other books out on loan, but less than 6. The system is updated.
- A use case description, a UML activity diagram, or a description in some formal language is associated with each use case.
- A use case represents a set of related requirements that start to emerge during initial requirements capture, and which are later refined in successive iterations.
- A use case may be associated with a collaboration and communication diagram and to demonstrate how specific scenarios are realised.

3. Additional Notation: << includes>>

- UML stereotype **<<include>>:** identify common behaviour in two or more of use cases, or when a component can be used to implement part of use case.

- Direction of arrow indicates that source use case makes use of target use case.

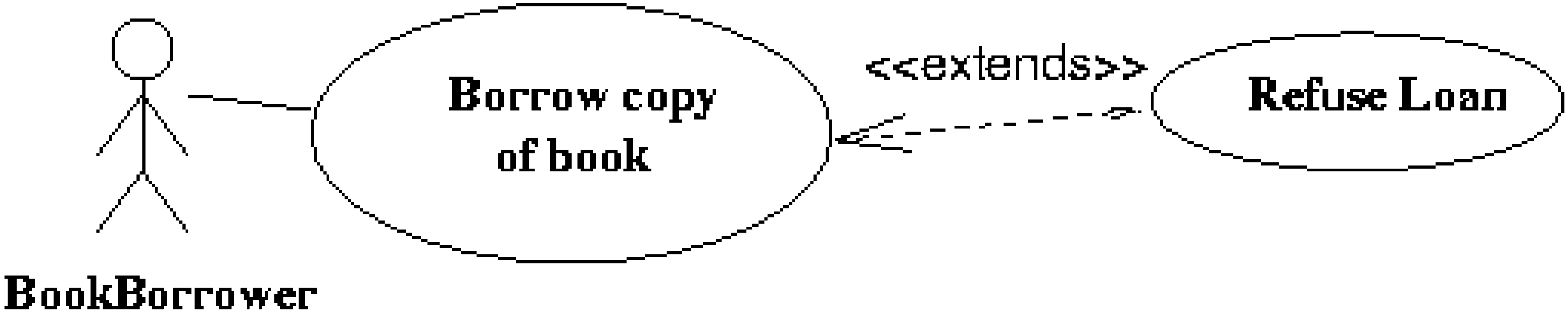


3. Additional Notation: <<includes>>

- Scenarios that are instances of the source use case contain sub scenarios that are instances of the target use case.
- Explicitly document target use case in use case description, example
- …. The system checks whether there is a reservation on the book *(use case Check for reservation*).
- Benefits: means of identifying possible reuse, which has further planning implications.
- Disadvantages:
  - By looking for reuse in the process oriented use case model, reverting to a top down functional decomposition. OOAD tries to avoid this.
  - More difficult for users to read diagrams.

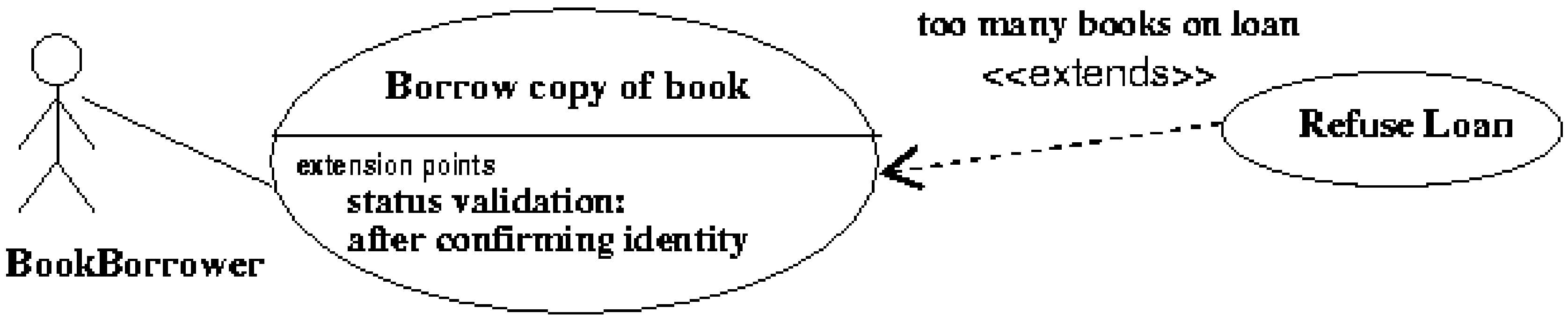
3. Additional Notation: <<extends>>

- If a use case incorporates two or more different scenarios, identify generic or main case.
- Other cases are variant or subsidiary cases.
- Use the UML stereotype **<<extends>>** arrow from the subsidiary case to the main case - opposite of <<uses>>.



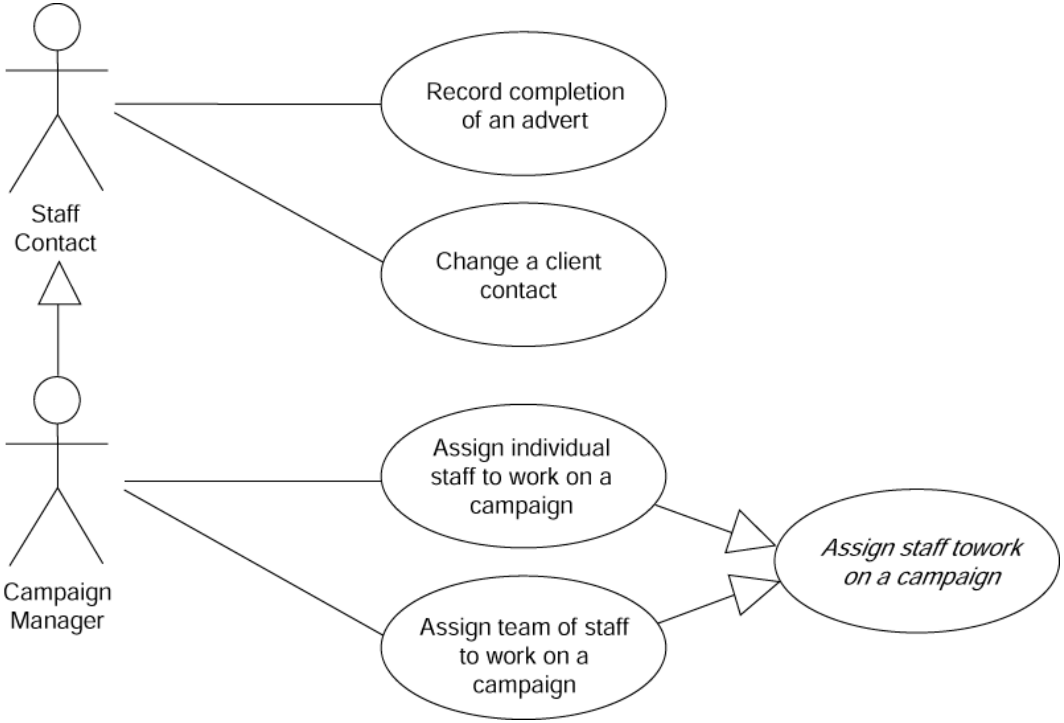
3. Additional Notation: <<extends>>

- In the new use case description must include:
  - The condition under which the exceptional case applies.
  - The point at which the condition is tested and the behaviour diverges, known as the extension point.



3. Additional Notation: Generalisation

- Generalisation can also be applied to:
  - Actors.
  - Use cases. Base use case is an abstract use case.


Record completion of an advert
Staff Contact
Change a client contact
Assign individual staff to work on a campaign
Assign staff towork on a campaign
Campaign Manager
Assign team of staff to work on a campaign

4. Actors

- Actors who derive benefit from a use case are called beneficiaries or primary actors.

- Actors that represent non-human users: Do we show interaction? Different possibilities (Fowler and Scott, 1997). Options:
  - Always.
  - When it is the other system that initiates contact.
  - When it is the other system that derives benefit from the contact.
- Some (few) systems analysts adopt the view that only humans should be represented as actors.
- Also, some discussion in texts as to whether secondary actors should be shown. Recommend their inclusion for clarity.

5. System Boundary

- The system boundary should be represented as a box around the use cases in a use case diagram, that separates the use cases from the actors.

- When drawing a use case diagram to represent a simple system, common to omit the box.

- UML guide specifies that there should be a single set of use cases and one system boundary per diagram.

6. Identifying Use Cases

- Identify the actors.
- For each actor, determine:
  - What they need from the system, use cases that have value for them – primary actor.
  - Any other interactions with the system, use cases in which they play a role but do not derive direct benefit – secondary actor.
- In addition to an understanding of task and users, must have:
  - Knowledge of risk associated with each use case.
  - A rough estimate of time required to implement each use case.

7. Planning

- Once set of use cases known that one is contracted to provide, prioritise them and allocate them to a specific iteration.

- Prioritise according to risk.

- System validation - walk the use case, also facilitates derivation of system tests.

8. Possible Problems with Use Cases

- Danger of developing a non object-oriented system with a poorly specified architecture and inadequate class modelling. Why?
  - Use refactoring before any new functionality added in a subsequent iteration.

- Confusing design with requirements.
  - Use cases should specify what happens and not computational sequence of steps (solutions) or a specific technology-based order.

- Missing requirements - use case engineering and conceptual class modelling in parallel.

9. Components

- Will have to adapt use case model to match available components. Architectural decision.
- When using a component that we did not develop, do we show it in our design diagram?
- Or treat component in the same fashion as a programming language construct i.e. like procedural or class libraries!
- Depends on complexity.
- Example: an intelligent query engine.
- Also note that there is an underlying assumption in use case diagrams that data infrastructure will be developed/composed to support persistence.

10. Reading

- Chapter 6 in Bennett et al. or
- Chapter 7 and 8 in Stevens and Pooley