CS4125
SYSTEMS ANALYSIS
SPRING SEMESTER 2010-2011

J.J. Collins
Dept of CSIS
University of Limerick

Lecture E (W3L2)

Overview of Use Case Engineering Part 1

Preface: Polymorphism

- Email on Monday Week 3 states that polymorphism reduces coding effort required.
- Handout from Fowler's text entitled Refactoring illustrates how polymorphism facilitates maintenance:
  - Following principle of "Finding what varies and encapsulating it"
  - Easier in Fowler's example to change rules for calculation of frequent renter points for Regular movie.
  - Focus on pages 13-17, and 44.
  - Will have a more indepth look later in semester.

1. Introduction

- Business analyst models:
  - Organisational structure
  - Workflows and processes
    - Both formal and informal
  - IT systems
- May also produce a high-level list of requirements.
- Business analyst must have:
  - Fluency in the application domain
  - Understand vertical (market sector) roadmap
    - For example: business analyst working for supply chain vendor
    - Sectors: pharmaceutical, medical device, finance, automotive, finance, etc.
  - High interaction competency

1. Introduction

- The Requirements Engineer responsible for specification of requirements.
- Using techniques and methodologies to elicit requirements:
  - SQIRO:
    - Sampling
    - Questionnaires
    - Interviews
    - Reading
    - Observation
  - Dynamic Systems Development Methodology
- All requirements should be evidence-based
  - Scientific approach to data capture whose outputs are derived using quantitative and qualitative methods from statistics.
- This is not an Requirements Engineering (RE) module!
  - Understanding 2 lectures of slides will not make you an expert if RE

1. Introduction

- Taxonomy of Requirements
  - Functional
  - Non-functional Requirements (NFRs)
    - See taxonomies (categories) in handouts
      - Example of a Volere template used to capture use case descriptions.
      - Example from Sommerville.
  - Usability
    - Not universally agreed as a separate category.

- EXERCISE: Find a taxonomy that you are happy with!

- NFRs can be broken down into 3 areas according to Gordon:
  - Quality attributes / architectural uses cases
    - Modifiability, extensibility, portability, performance, etc.
    - Sometimes collectively referred to as the “ilities”.
  - Business
    - i.e. Must be Sarbannes-Oxley / FDA / some other standard compliant
  - Technical
    - Must develop in Java and use Websphere Message Broker

Ian Gorton. Essential Software Architecture. Springer. 2006

2. Overview of Use Cases

- Requirements model consists of
  - Use Case diagram (UCD)
  - Use Case description for each use case
  - Documentations on NFRs
  - Prototypes
- Use cases document the behaviour of the system from a user's perspective.
- Use case modelling used for:
  - Requirements capture.
  - Planning iterations.
  - Validating system.

2. Overview of Use Cases

- A use case represents a task (coherent unit of functionality), which has to be done with support from the system under development.
- An actor represents a role played by a user of the system. User implies anything external to the system.
- A line connects an actor to a use case if the actor may take part in the task.
- A use case diagram represents sets of things and possible interactions, rather than individual things and definite interactions (classes and objects).
- An instantiation of a use case is a scenario, and is documented in the textual description.

3. UML Modelling Example

The Problem

- You have been contracted to develop a computer system for a university library. Currently based on a legacy system and card index for browsing.

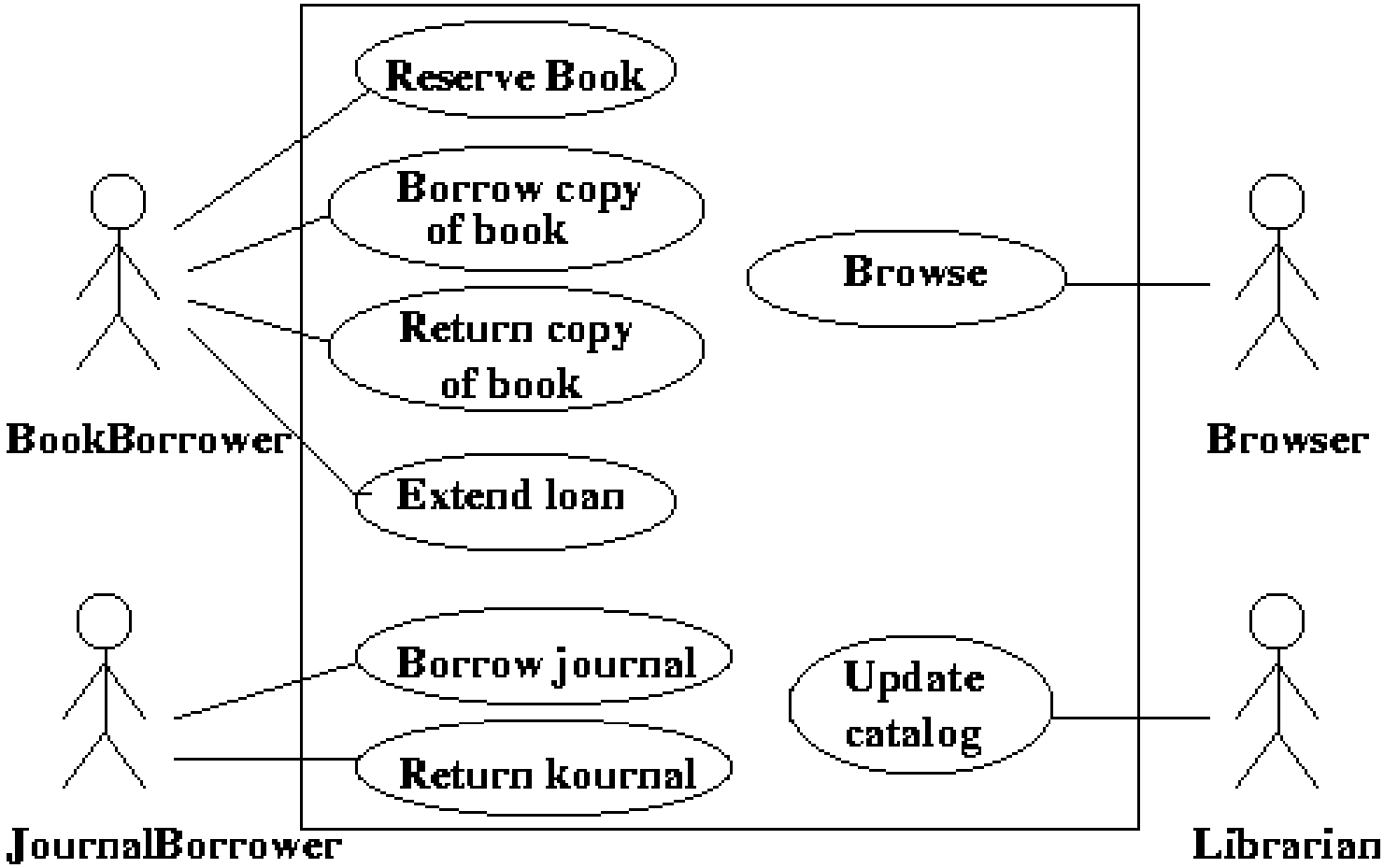
- Following facts:
  - Books and journals: may have several copies of a given book. Library members can borrow up to 6 books at a time. Staff members may borrow up to 12 books and borrow journals.
  - Borrowing: keep track of borrowing and returning. Possible extension - generate reminders.
  - Browsing.

3. UML Modelling Example

Requirements: Use Case Model

- Use Case Model: user-oriented approach to systems analysis.
- An actor is a user of the system in a particular role.
- A use case is a task that an actor needs to perform with the help of the system, e.g. Borrow copy of book.
- Name use case after what happens in the generic case.
  - Verb-based
- For each use case, document sequence of steps as a use case description.

3. UML Modelling Example

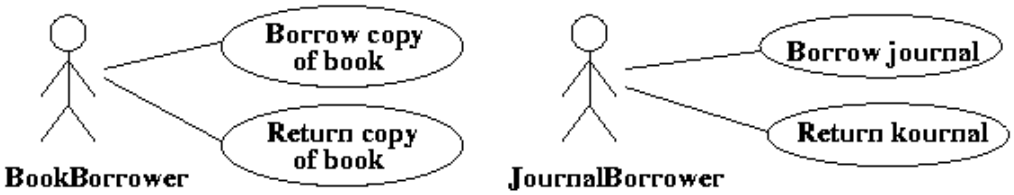


Borrow copy of book: A BookBorrower presents a book. The system checks that the potential borrower is a member of the library.....

3. UML Modelling Example

Software Development Process: Scope and Iterations

- Single release approach to systems development has history of failures.
- Instead, several iterations through development process.
- First iteration: delivery of system with very basic functionality.
- An iteration can deliver enough of the system to allow certain use case to be followed out, but not others.
- First iteration: develop following use cases -
  - Borrow a copy of book
  - Return copy of book
  - Borrow a journal
  - Return journal.
- In a real project, might have to diagram and describe all use cases in first iteration, serves as a contractual basis.



3. UML Modelling Example

Analysis: Identifying Classes

- Systems analyst identifies candidate classes in the analysis phase based on Use Case descriptions
- Identify the key domain abstractions.
  - Abstractions equal classes.
- Use noun identification technique applied to requirements.
  - Underline nouns and noun phrases to derive a list of candidate classes.
- Data driven design (DDD) versus responsibility driven design (RDD).

3. UML Modelling Example

- **Books and Journals:** The library contains books and journals. It may have several copies of a given book. Some of the books are for short term loans only. All other books may be borrowed by any library member for three weeks. Members of the library can normally borrow up to six items at a time, but members of staff may borrow up to 12 items at one time. Only members of staff may borrow journals.
- **Borrowing:** the system must keep track of when books and journals are borrowed and returned, enforcing the rules described above.

3. UML Modelling Example

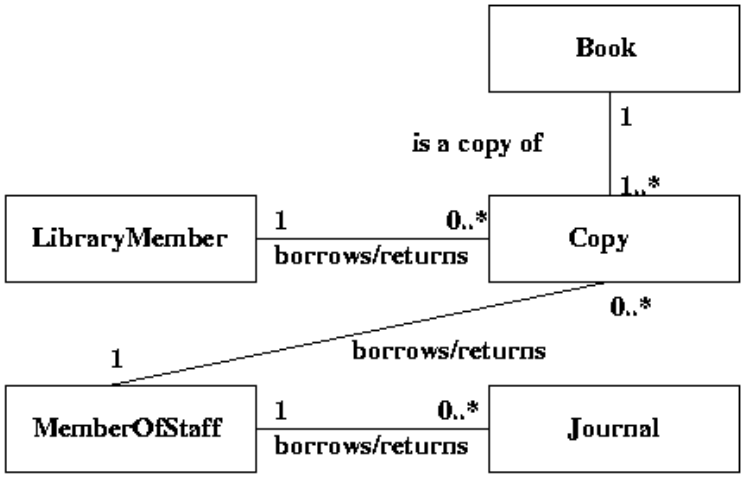
Heuristics (rules of thumb) used to eliminate poor candidates in the analysis phase.

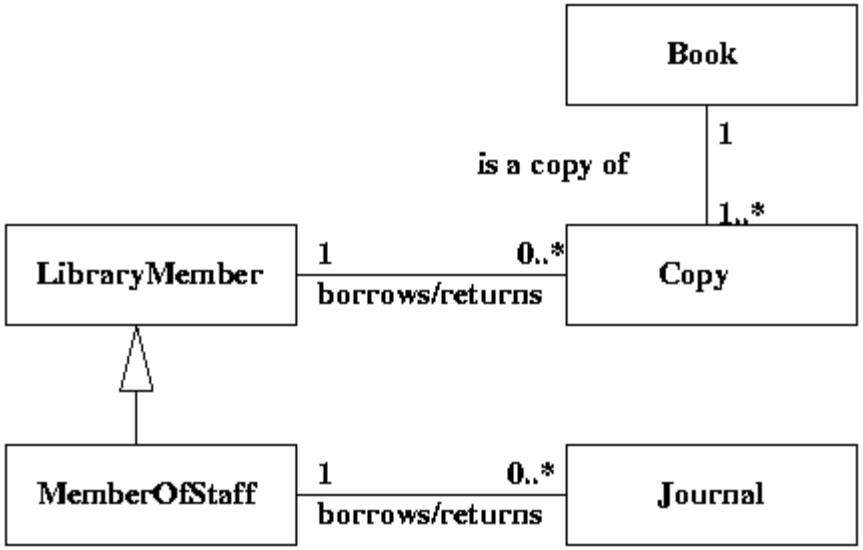
- Discard those that are not good candidates.
  - library: because it is outside the scope of our system.
  - short term loan: an event, not a useful object.
  - member of the library: same as library member.
  - week: measure of time, not a thing.
  - item: vague.
  - time: outside scope of system
  - system: part of meta-language of requirements description.
  - rules: same reason.
- The following classes are left:
  - book
  - journal
  - copy (of book)
  - library member
  - member of staff

3. UML Modelling Example

Analysis phase class diagram – first cut!

- Why - to clarify understanding of domain, to sanity-check the coupling.
- Use a UML class model diagram to illustrate associations.
- Shows multiplicity of the associations.
- Nothing stated about navigability of the associations.
- MemberOfStaff share same associations that LibraryMember does. Revise class model.
- Generalisation between MemberOfStaff and LibraryMember.





5. Use case Descriptions

- Work through example from CSE in Dublin.
  - Reprinted with kind permission.
- Look again at Volere template.

6. Reading

- Bennett, McRobb, and Farmer: chapter 6.