

Use Cases

G. Falquet, L. Nerima

References

Jacobson, M. Christerson, P. Jonsson, G. Overgard
" Object-Oriented software engineering : A use case driven approach " Addison-Wesley 1992

Jacobson, M. Griss, P. Jonsson
" Software Reuse : Architecture, Process and Organization for Business Success " Addison-Wesley 1997

ISI

June 2001 - G. Falquet, L. Nerima Use Cases 2

Motivation (software engineering)

" Solve the right problem "

- ❖ Requirement analysis
 - ❖ Determine the requirements - what the system should do
 - ❖ Understand the requirements
 - ❖ Delimit the system
- ❖ User centred analysis
 - ❖ Capture requirements from the user's point of view
 - ❖ Integrate user views

Motivation (business reengineering)

- ❖ Consider the organisation as a system
- ❖ Understand what the organisation does
- ❖ Reconfigure the organisation

Reference

- ❖ I. Jacobson. *The Object Advantage, Business Process Reengineering with Object Technology*. ACM Press.

UML Development phases

- ❖ Inception
 - Find the limits of the system, specification, feasibility
 - ❖ **output**: requirement model = use cases
- ❖ Elaboration
 - Analysis, architecture, resources, planning
 - ❖ extend use cases
- ❖ Construction
 - Design, implementation, tests
- ❖ Transition
 - Delivery, maintenance

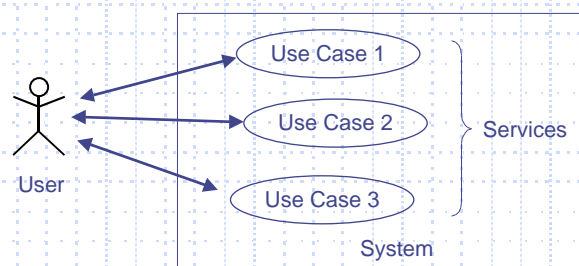
Use cases (principles)

- ❖ What the system should do (desired behaviour)
- ❖ Not how to carry out that behaviour
 - ❖ No programming details
 - ❖ Independent of the realisation
- ❖ A tool to communicate
 - ❖ end user / domain expert <---> developer / designer

Use cases and User-System Interactions

- ❖ A use case describes the **interaction** between an **actor** and the system
- ❖ The interaction yields a **tangible result** for the actor
- ❖ The interaction is a **sequence of actions**
- ❖ Transactions are expressed in terms of **domain objects**

Users and the System

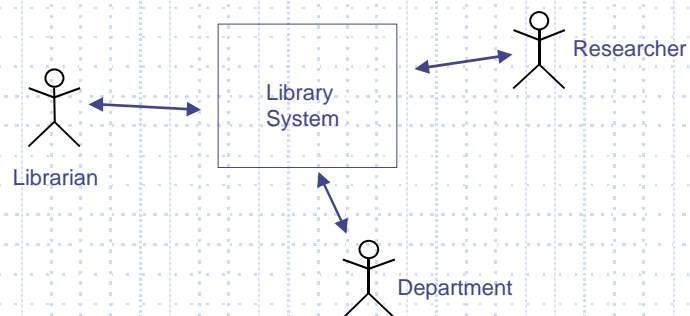


Actor

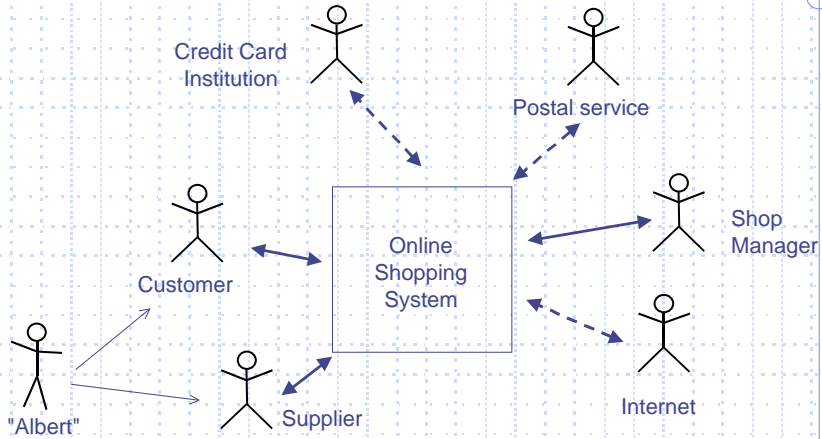


- ❖ What exists **outside** of the system
- ❖ Anything that needs to **exchange information** with the system
 - ❖ person, machine, organisation, other computers, other system
- ❖ Corresponds to a generic **role** that a user plays
 - = a way of using the system
- ❖ The same person (machine, ...) may play several roles

Example: Library



Example

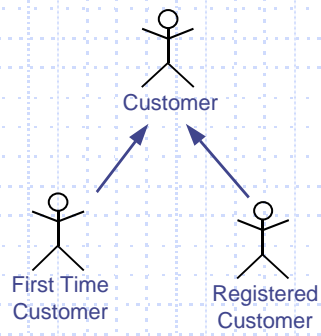


June 2001 - G. Falquet, L. Nerima

Use Cases

11

Classification of Actors



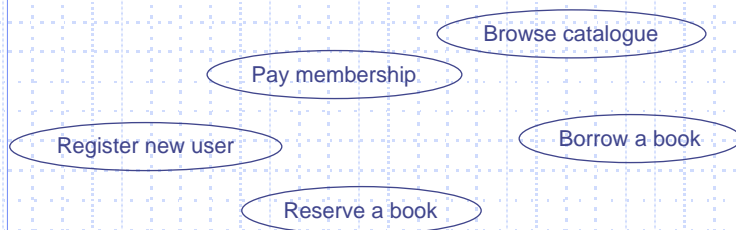
June 2001 - G. Falquet, L. Nerima

Use Cases

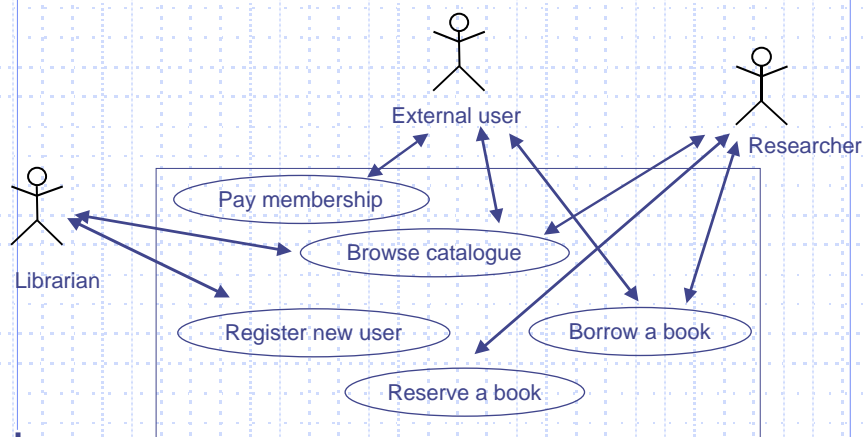
12

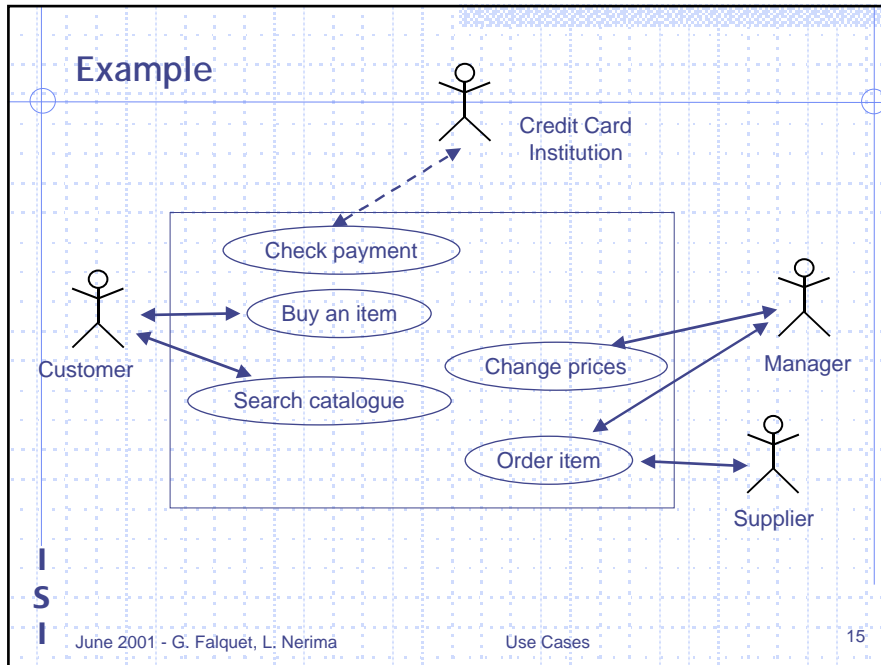
Use case: definition

- ❖ Description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor.



Use cases and their users (actors)





Use cases and scenarios

Use case: **Search library catalogue**

- ❖ **Main flow of events:** The user clicks the 'search' button. The system displays a text field in which the user can enter one or more keywords. The user validates the entry by clicking the 'go' button. The system displays a list of relevant items.
- ❖ **Exceptional flow of events:** The user opens a terminal window. The user types 'sr', a list of keywords, and '<enter>'. The system displays a list of relevant items.

June 2001 - G. Falquet, L. Nerima Use Cases 16

Scenario

- ❖ Describe the flow of events
- ❖ In natural language
 - ❖ in terms of domain objects (-> object diagrams)
 - ❖ and actions
- ❖ One main scenario and alternative scenarios

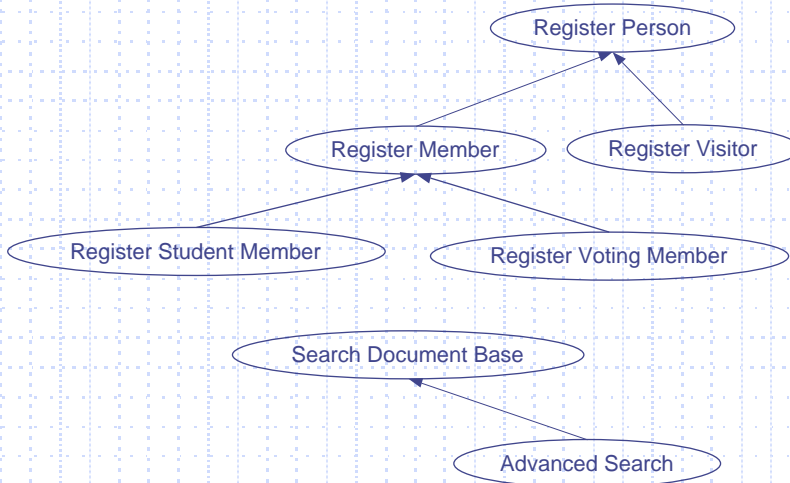
Library Example: Locate Media

- ❖ Primary Path: (Media Located) The librarian and borrower (researchers) will both search for media topics in the library through a computer display. The topics can be located by type, title, author, keywords in the description as well as specific media type selections (ex software format). The results of the search are displayed to the researcher in summary format. The detailed descriptions of each media topic can then be displayed as well as printed.
- ❖ Alternate Path: (Media not Located) The system does not locate the media. The researcher is informed of this result and can then edit the previous query or cancel the request.

Use case: Add Media

- ❖ Primary Path: (Add Media for New Topic) The librarian will interact with a computer display to enter in a new media topic and copy. The librarian will enter the title, author, and description for the media topic and serial numbers for each new copy. The librarian will also be able to enter the following information for specific types of media topic:
 - ❖ Books: pages
 - ❖ Software: format and version
 - ❖ Video Tapes: format and ratingThe new media is entered into the library system and is now available for borrowers to search for and check-out.
- ❖ Alternate Path: (Add Media for Existing Topic) The librarian will search for the media topic by title and author and add the new serial numbers for the additional copies.

Use case taxonomy



Taxonomy (cont.)

- ❖ Generic - specific relationship.
- ❖ Extract common behaviour.
- ❖ Behaviour of the child must include the behaviour of the parent (substitutability).

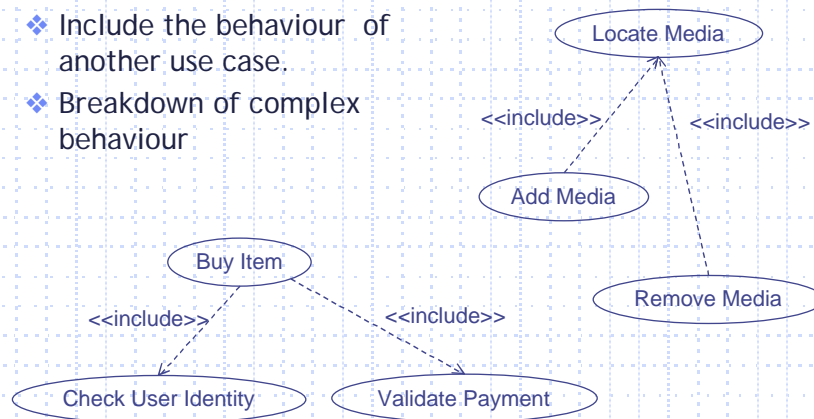
Register Member
Type name
Obtain member no.

Register Student Member
Type name
Type university name
Type student no.
Obtain member no.
Update student database

I
S
I

Use Case Relationships : Inclusion

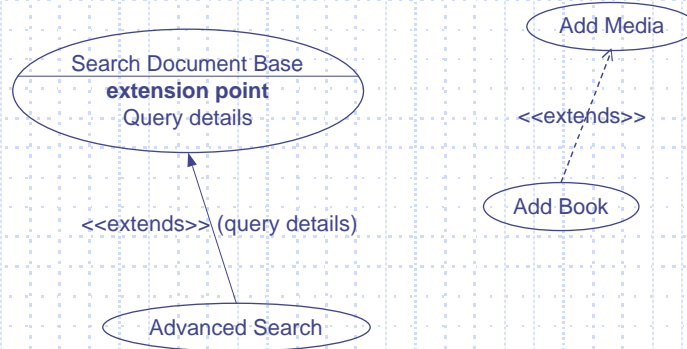
- ❖ Include the behaviour of another use case.
- ❖ Breakdown of complex behaviour



I
S
I

Use Case Relationships : Extension

- ❖ Additional sub-course of events (conditional)
- ❖ Complex alternate path at given extension points

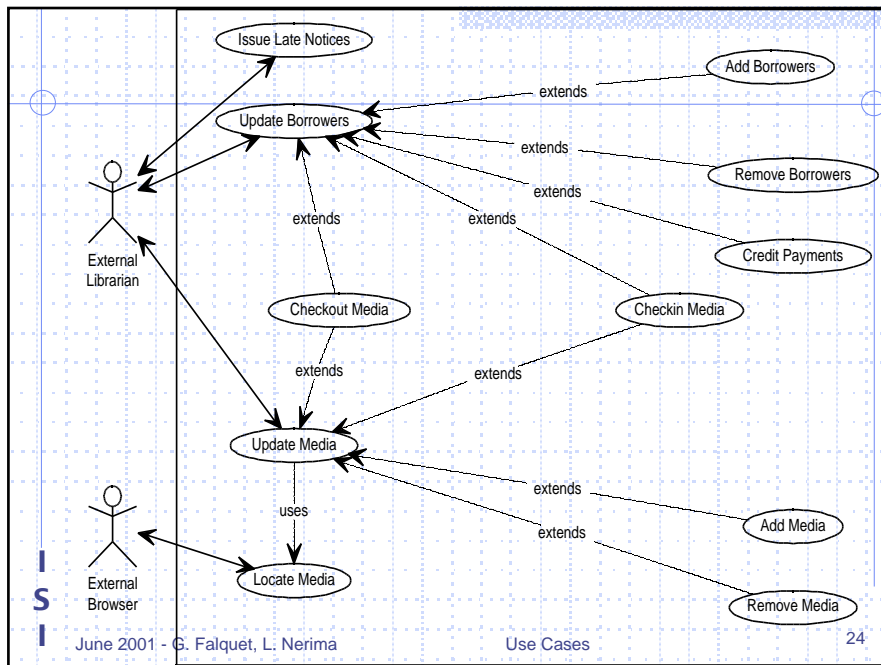


I
S
I

June 2001 - G. Falquet, L. Nerima

Use Cases

23



I
S
I

June 2001 - G. Falquet, L. Nerima

Use Cases

24

Scenario Writing Rules

- ❖ Describe the activity : "what" not "how"
E.g. operator checks user identity
- ❖ Keep it simple
Decompose complex cases with <<uses>> or <<extends>>
- ❖ Autonomy
Do not mix use cases
- ❖ Direct style
No ambiguity, no approximation ("very", "rather", "somewhat", "few", "often", "in general")
- ❖ A scenario is a (long) transaction
Beginning and end
All or nothing executed

I
S
I

Scenario Writing: Use Linguistic Constructs

while condition do :

```
<< action >>  
<< action >>  
<< action >>
```

if condition:

```
then << actions >>  
else << actions >>
```

repeat n times :

```
<< actions >>  
<< action >>  
<< action >>
```

choices :

```
condition : << actions >>  
condition : << actions >>  
condition : << actions >>  
....
```

I
S
I

Scenario: Check user identity

user enters desired amount with the keyboard
user presses 'OK'
check if amount realizable with available banknotes
while amount not realizable **do** :
 system proposes another amount
 user may change the amount
 user presses 'OK'
prepare banknotes
eject user card
wait for user to take banknotes

I
S
I

June 2001 - G. Falquet, L. Nerima

Use Cases

27

Use Cases and Objects

- ❖ Two different modelling axis
 - ❖ Different views of the system
- ❖ Relationships between use cases and objects
 - ❖ Objects represent
 - ❖ things used during a flow of events
 - ❖ tasks performed
 - ❖ communications with the world
 - ❖ Objects formalise the internal organisation of use cases

I
S
I

June 2001 - G. Falquet, L. Nerima

Use Cases

28

Objet Modelling

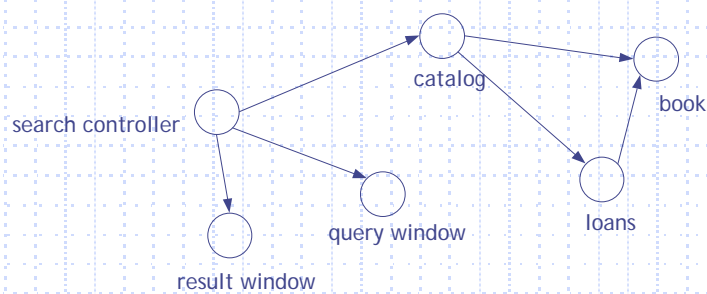
- ❖ Represent real-world entities (concrete or abstract)
 - ❖ a book owned by the library
 - ❖ a borrower
 - ❖ a loan
 - ❖ a user interface window
 - ❖ an event's occurrence (button 'OK' was pressed at 16:22.33.01)
- ❖ Contain data
- ❖ Have a state and a behaviour
- ❖ May be related to each other

- ❖ Objects are in the system

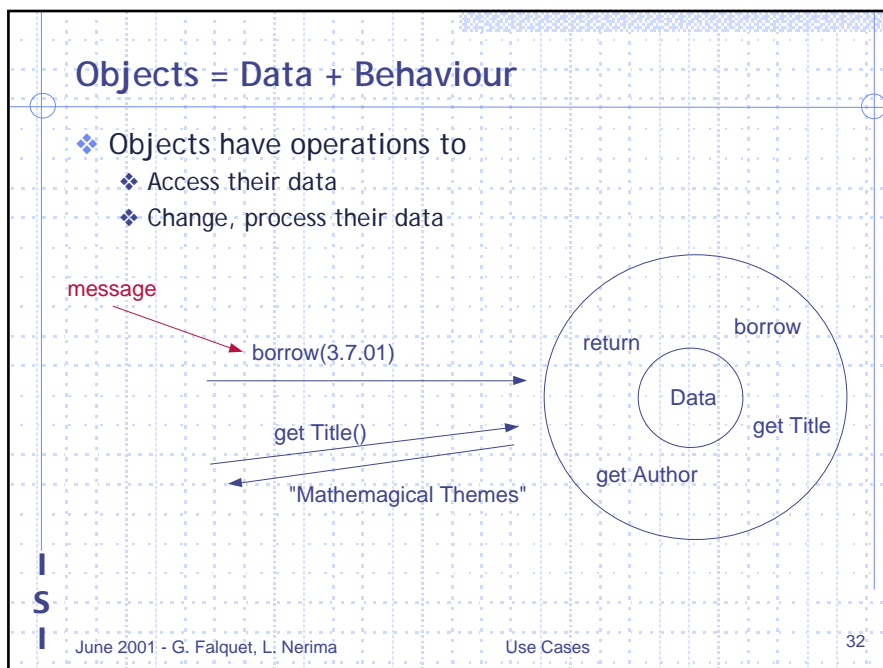
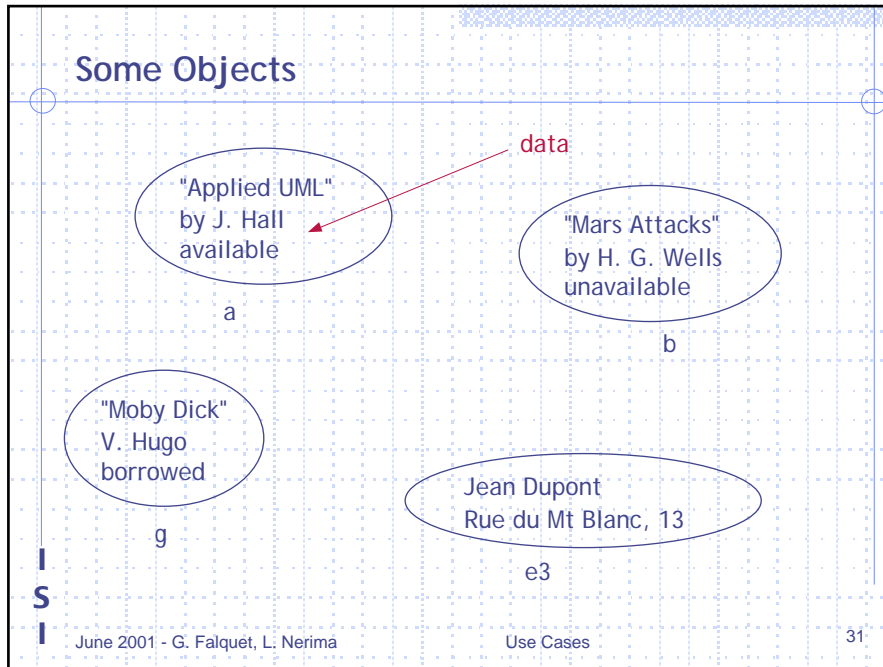
I
S
I

System = Set of Interacting Objects

- ❖ Objects
 - ❖ exchange messages to execute operations
 - ❖ form the memory of the system

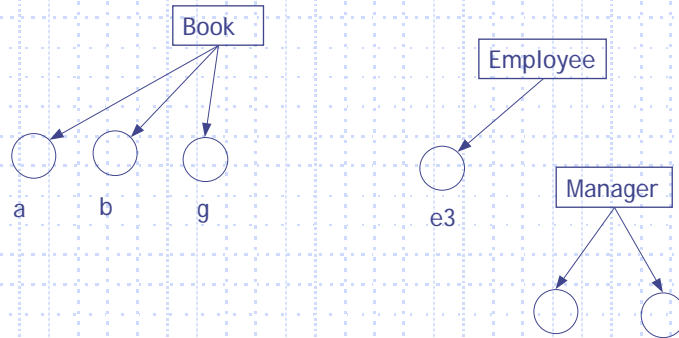


I
S
I



Classes

- ❖ Categorise objects



I
S
I

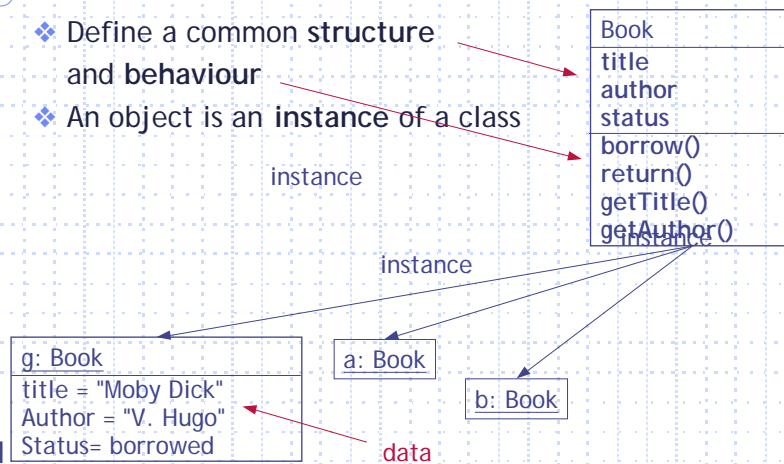
June 2001 - G. Falquet, L. Nerima

Use Cases

33

Classes and Object Structure

- ❖ Define a common structure and behaviour
- ❖ An object is an instance of a class



I
S
I

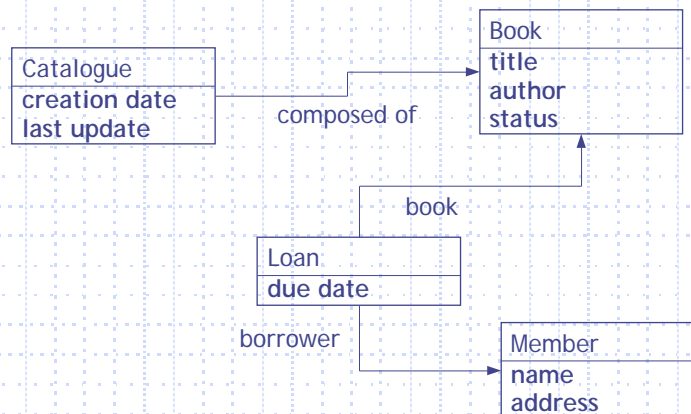
June 2001 - G. Falquet, L. Nerima

Use Cases

Book b

34

Classes have Relationships



Represent Which Entities ?

- ❖ Use cases determine entities to be represented.

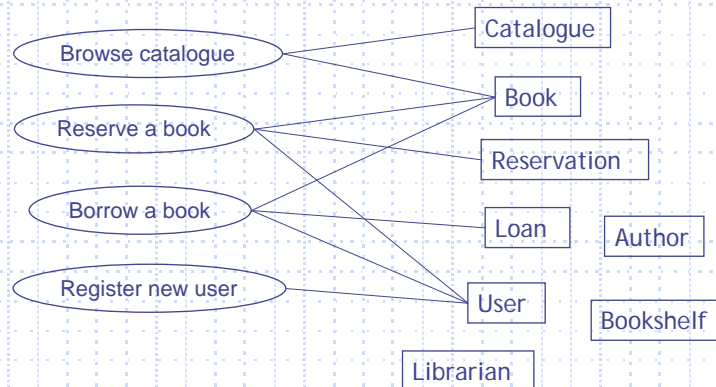
Scenario: "Check the **book's** availability. Check the **user's** status; Register the **loan**"

=> The system needs objects to represent **books**, **users** and **loans** (not bookshelves, doors, seats, tables, ...)

- ❖ Avoid the endless modelling syndrome.

Use Cases and Objects

- ❖ Refinement of a use case --> implementation
- ❖ Connect use cases to relevant object classes



I
S
I

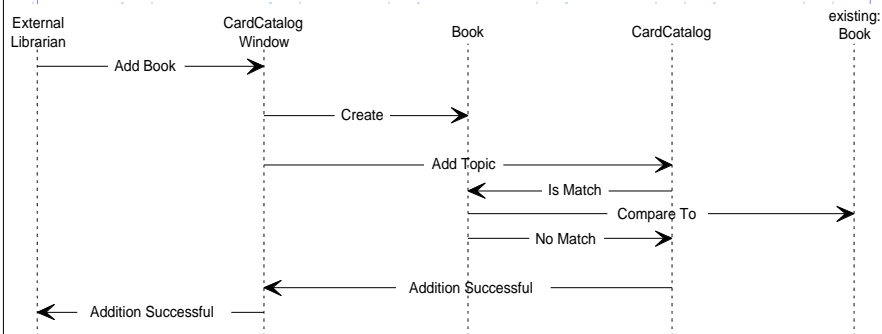
June 2001 - G. Falquet, L. Nerima

Use Cases

37

Scenarios and Objects: Sequence Diagrams

The librarian will add a new book to the library by creating a new book media topic through a card catalog window. In creating the media copy (book), the librarian will supply parameters of title, description, author, and pages. An existing topic is searched for in the existing topics. The successful addition is reported to the librarian. The librarian can then add a new media copy for each copy of the book (refer to Add New MediaCopy).



I
S
I

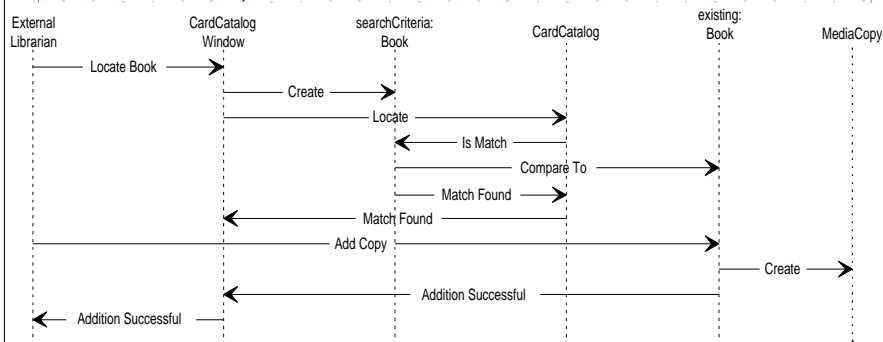
June 2001 - G. Falquet, L. Nerima

Use Cases

38

Scenario: Add Media Copy




The librarian first requests the media topic from the card catalog window. A set of search criteria is filled in and used to locate the media topic in the card catalog. Once found, the media topic can have many operations performed on it. In this instance, the librarian requests that a new copy be added. The librarian will supply a bar code for the new copy. The media topic adds the new copy and the librarian is informed of the successful completion.



UML for Business Reengineering

- ❖ System = Organisation (company, office, etc.)
- ❖ Use case = Generic service
- ❖ Object = Parts of the system
- ❖ Objective: understand how the system works
- ❖ Eventually: change the system / organisation

Object Categories

- ❖ Entity objects 
 - ❖ Things and products used during a flow of events (scenario)
- ❖ Processors 
 - ❖ Tasks that must be performed
- ❖ Interfaces 
 - ❖ Tasks that communicate with the outside world

Example: Restaurant

