

Sequence Diagrams

Ferd van Odenhoven

Fontys Hogeschool voor Techniek en Logistiek

May 27, 2015

Sequence Diagram

Definition

A sequence diagram displays how a use case could evolve using messages between objects / participants.

The messages follow the relations between classes in the class diagram.

- Typically each concrete scenario has its own sequence diagram.
- If the domain changes, the sequence diagram might have to be altered too!

Sequence Diagram Elements

In UML1 the messages are between *objects*, as of UML2 they are termed more general as *participants*. In the figure below the participant is an object.

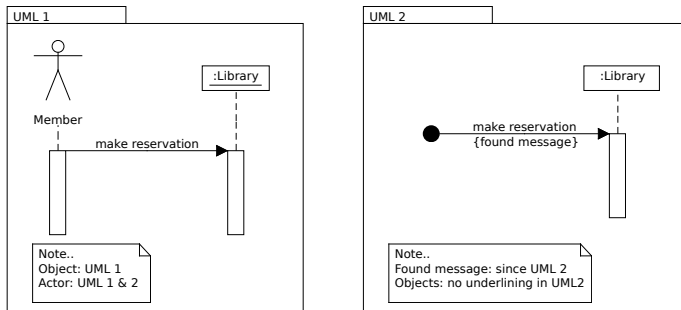


Figure : Message with actor (uml 1 & 2) or 'found message' (uml 2)

Sequence Diagram Elements

Participants can be anonymous objects or named objects. On the life line, a bar denotes that a participant is active

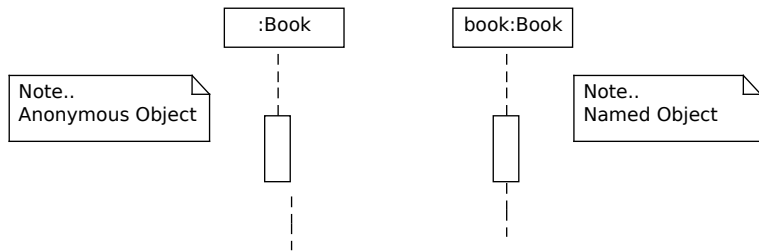


Figure : Objects are the participants here.

Sequence Diagram Elements

Messages may or may not (if the diagram is unambiguous) have return values. A return message is drawn as a dashed arrow. 'Self calls' are messages to itself (`this`) object.

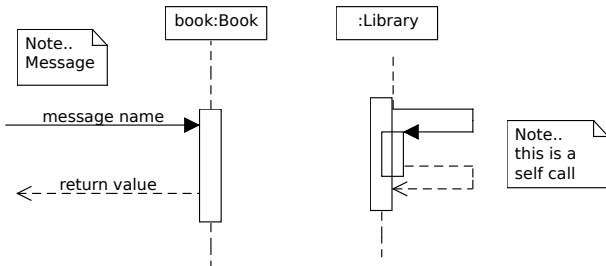


Figure : Messages with returns and self calls.

Sequence Diagram Elements

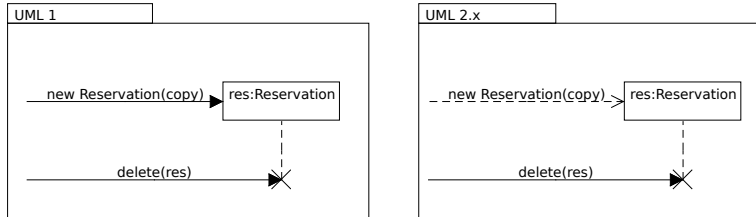


Figure : Construction and destruction of Objects.

In case of garbage collection there is no destructor method!
Showing only the cross symbol shows that the object is no longer present.

CRC-cards

A CRC-card looks as follows :

Class:	Nr:
Responsibilities:	Collaborations:

Table : CRC-cards can help in dividing the responsibilities over the relevant classes.

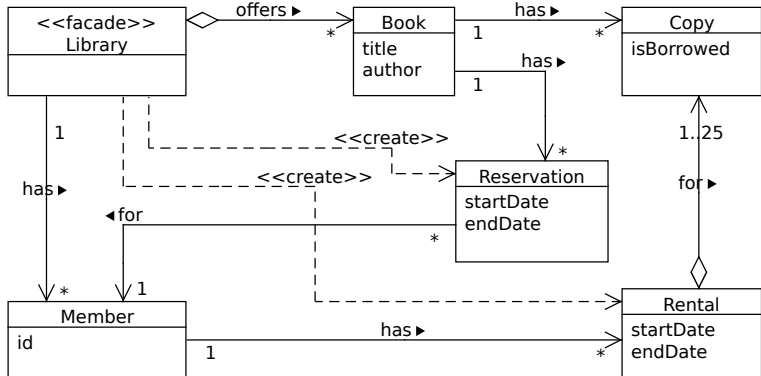
Why use CRC-cards for sequence diagrams?

- A sequence diagram starts at the right object: the one that has the responsibility to start this use case.
- If for the next step not this object but another object is responsible, a CRC-cards could provide this information.
- If the sequence diagram looks awkward: too complicated or messy, redistribute certain responsibilities or change the domain model, alter CRC-cards and the new sequence diagram should be better.

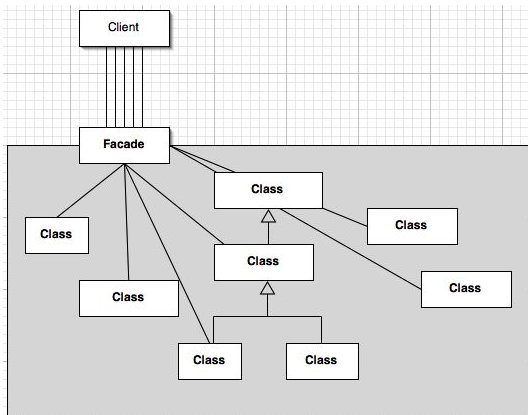
Library Example: Usecase

Use Case:	Make book lending
Actor:	Member m1 of the library
Precondition:	Actor is logged on
Description:	<ol style="list-style-type: none">1. Actor searches for book with data: title 'The Ring 1' and author 'Tolkien'2. Actor activates 'make rental'3. System checks if there is a copy available4. A copy is available and is set to borrowed5. A new rental has been created6. New rental is added to rentals of member m17. System returns a message: "Rental creation OK."
Extensions:	
Exceptions:	

Library Class Diagram



Facade pattern



CRC-card Library

Class: Library	Nr: 1
Responsibilities: <ul style="list-style-type: none">- Manage accounts and members- Manage book catalog- Create lending- Create and make reservations	Collaborations: <ul style="list-style-type: none">AccountMemberBookCopyRentalReservation

Table : CRC-card for the Library class

CRC-card Book

Class: Book	Nr: 2
Responsibilities: <ul style="list-style-type: none">- Manage Reservations for this book- Find member who has a reservation for this book- Add and remove reservations	Collaborations: Reservation Member Book

Table : CRC-card for the Book class

CRC-card Member

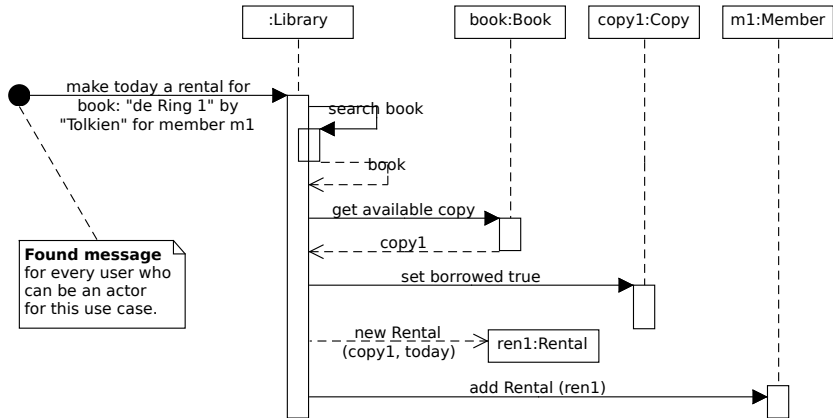
Class: Member	Nr: 3
Responsibilities: - Manage Rentals for this member	Collaborations: Rental

Table : CRC-card for the Member class

Class: Reservation	Nr: 4
Responsibilities: - Store member that has this reservation	Collaborations: Member

Table : CRC-card for the Reservation class

Library Sequence Diagram



Object diagram after creating rental

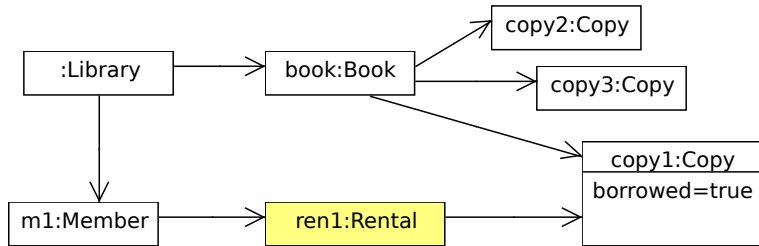


Figure : The object **ren1** and the connections to and from this object are new. The library has three copies of the corresponding book.

Train company

- Create a first version of a class diagram by introducing a facade class with name : TrainCompany
- Choose two use-cases you made earlier.
- Make sequence diagrams for these use cases.
- Do not add superfluous much details! Only the essentials to understand the sequence of steps.
- Improve your domain model if necessary and comment on your changes.