

Sample Designs

Massimo Felici

JCMB-1402 0131 650 5899

1BP-G04 0131 650 4408

mfelici@inf.ed.ac.uk

The SEOC process: Where are we?

1. Gathering Requirements

- Writing a Requirements Specification Document (e.g., see the VOLERE template)

2. Capturing functional requirements into Use Cases

- Describe use cases by a Use Case Template

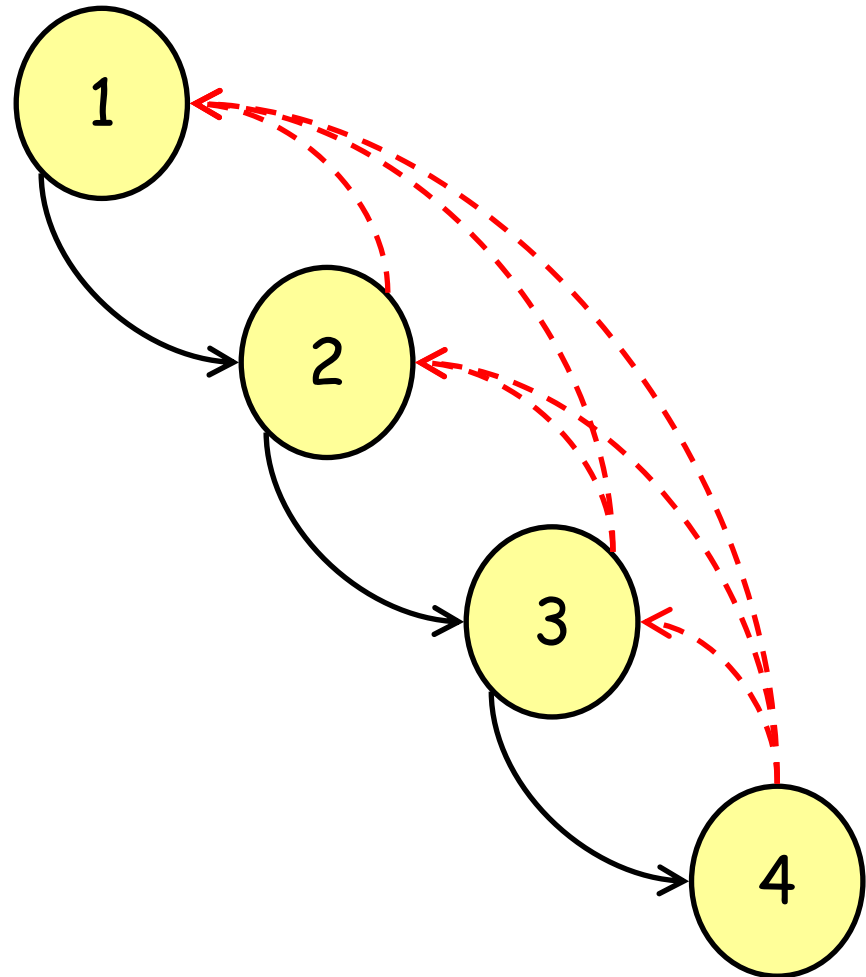
3. Modelling a preliminary system design into Class Diagrams

4. Validating your design by CRC cards



The SEOC process: Where are we?

1. Requirements
2. Use Cases
3. Class Diagrams
4. Validation

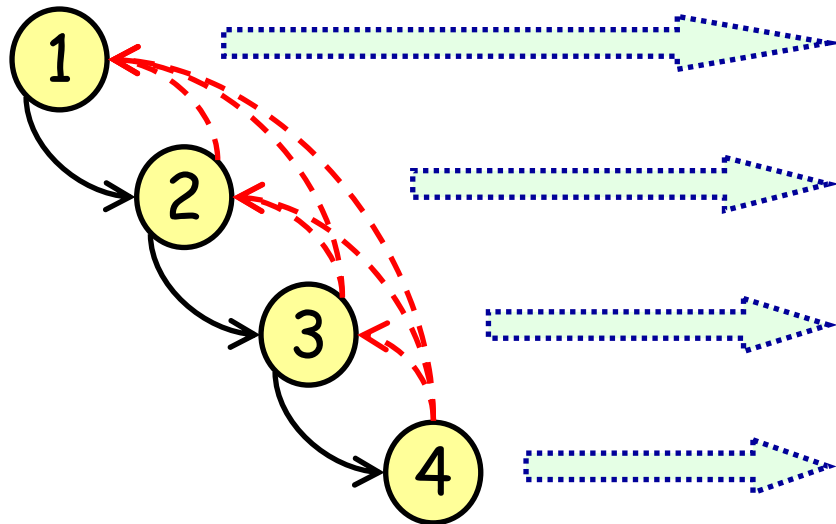


The SEOC Project Deliverable 1

1. Requirements Specification
2. Use Case Model
3. Class Model
4. Validation of Class Model
5. Deliverable Assessment

SEOC Activity Deliverables

The SEOC Process




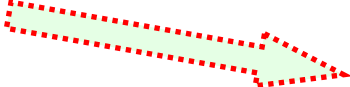
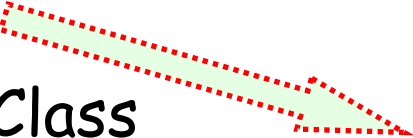
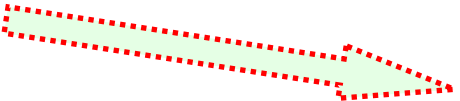
Deliverable 1

- Requirements Specification
- Use Case Model
- Class Model
- Validation of Class Model
- Deliverable Assessment

Deliverable 1 Assessment

Part 3 - Deliverable Marking Scheme

Deliverable 1

- Requirements Specification 
- Use Case Model 
- Class Model 
- Validation of Class Model 

Deliverable Marking Scheme		
Deliverable Part	Questions	Marks
Requirements Marks Limit: [20/100]	<p>Q1. Did you organise/collect the system requirements by using a Requirements Specification template (e.g., Volere)? Assess the quality of your Software Requirements Specification (SRS) document.</p> <p>Q2. Did you distinguish different types of requirements (e.g., functional or non-functional)? Assess how your SRS identifies different types of requirements.</p> <p>Q3. Do you believe you got most of the system requirements (requirements completeness)? Assess the extent to which you have elicited and gathered requirements from the main sources.</p> <p>Q4. Have you identified/resolved conflicting requirements (requirements correctness)? Assess the extent to which you have resolved conflicting requirements among different types (e.g., functional and non-functional) or across teams.</p>	[/ 5] [/ 5] [/ 5] [/ 5]
Use Cases Marks Limit: [30/100]	<p>Q5. Did you graphically represent the functional requirements by Use Cases? Assess to which extent your use case diagram captures main system functionalities and actors.</p> <p>Q6. Did you refine the use cases by generalization, include or extend relationships? Assess to which extent you have refined and structured use cases.</p> <p>Q7. Did you use a template for describing use cases? Assess to which extent you have clarified and described use case information (completeness and correctness).</p>	[/ 10] [/ 10] [/ 10]
Class Diagrams Marks Limit: [30/100]	<p>Q8. Does your class diagram identify the main classes of the system? Assess to which extent your class diagram realizes system use cases.</p> <p>Q9. Did you specify Attributes and Operations for each class? Assess the completeness of class specification.</p> <p>Q10. Did you identify relationships (i.e., Dependency, Association, Aggregation, Composition and Inheritance or Generalization) between classes? Assess the object orientation quality of your class diagram.</p>	[/ 10] [/ 10] [/ 10]
CRC Cards Marks Limit: [20/100]	<p>Q11. Did you construct CRC cards for your system design? Assess the completeness and correctness of CRC cards.</p> <p>Q12. Did you verify your Class Diagrams? Did you play any use case with the CRC Cards in order to verify your class diagram? Assess the quality and the coverage of your requirements and design verification by CRC cards.</p>	[/ 10] [/ 10]
Deliverable Mark		[/100]



CARMATCH

Sample Designs

CARMATCH Background

- CARMATCH is a franchising company that is being set up to promote car sharing
- Organizational goal: reduce carbon emissions
- CARMATCH seeks to promote car sharing
 - Matching potential car sharers
- CARMATCH consists of a three layer structure: (non-for-profit trust) global operation; national central operating company; local franchises
- In some countries, it offers insurances
- Main Profits: membership fees, consultancies, insurance commissions
- CARMATCH needs (has the requirements for) a computer system that can be used by its franchisees

CARMATCH Requirements

1. To develop a system that will hold information about members of the CARMATCH scheme
2. To match members up with other members as car sharers
3. To record insurance sales
4. To record details of potential and actual consultancy in the area of operation
5. The system must be capable of future expansion to incorporate information about toll and road-pricing and equipment sold to and installed for members

CARMATCH Requirements Specification

The System
Requirements Specification
Version ...

Table of Contents

PROJECT DRIVERS

1. The Purpose of the Project
2. Client, Customer and other Stakeholders
3. Users of the Product

PROJECT CONSTRAINTS

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

FUNCTIONAL REQUIREMENTS

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

NON-FUNCTIONAL REQUIREMENTS

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

PROJECT ISSUES

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Project Drivers

- CARMATCH background
- CARMATCH organization, Local governments, EU?, Local franchises, car sharers, etc.
- System Users?

Project Constraints

- Budget, Deadlines, Laws, etc.?

Functional Requirements

1. To hold information about members

1. 1
2. 2
3. 3

2. To match car sharers

3. To record insurance sales

4. To record details consultancies

Non-functional Requirements

Project Issues

Specification prepared by Date



Requirements Specification

CARMATCH Requirements Specification

The System
Requirements Specification
Version ...

Table of Contents

PROJECT DRIVERS

1. The Purpose of the Project
2. Client, Customer and other Stakeholders
3. Users of the Product

PROJECT CONSTRAINTS

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

FUNCTIONAL REQUIREMENTS

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

NON-FUNCTIONAL REQUIREMENTS

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

PROJECT ISSUES

18. Open Issues
19. Off-the-shelf Solutions
20. New Problems
21. Tasks
22. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

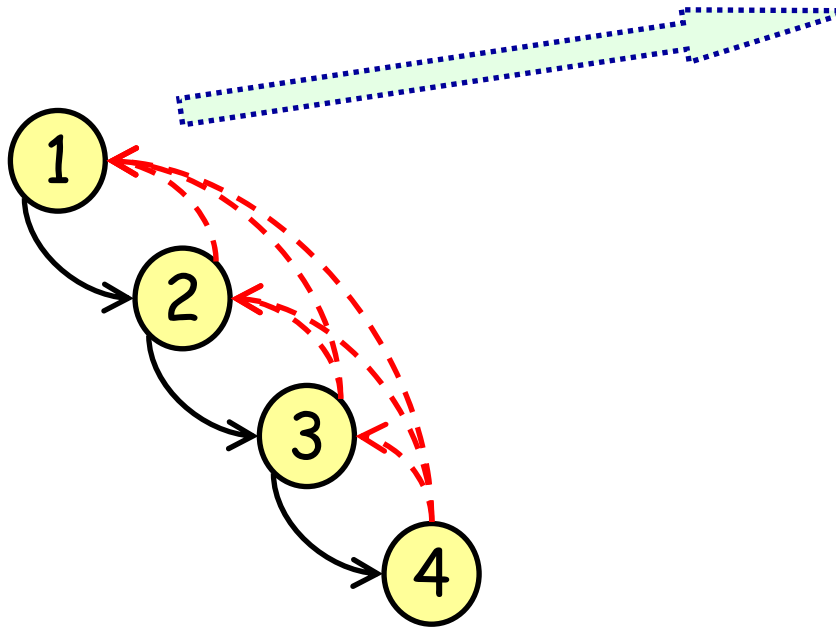
Specification prepared by Date

© 2004-2007

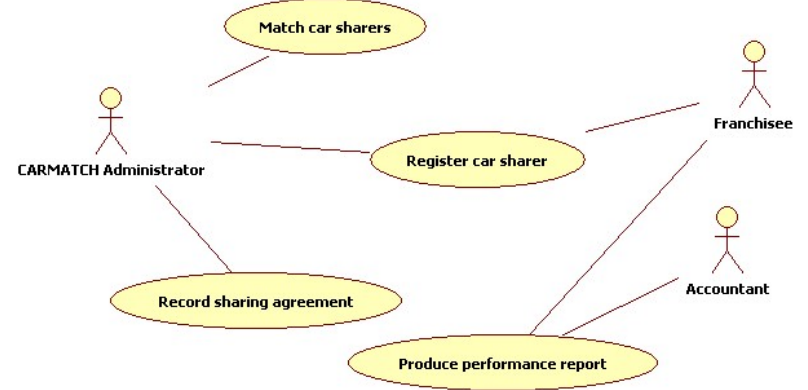
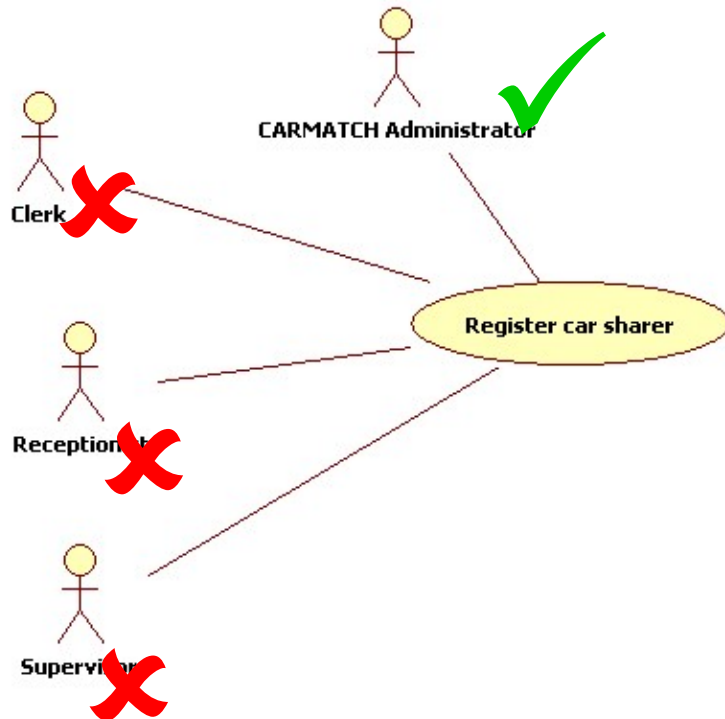
SEOC - Lecture Note 06

10

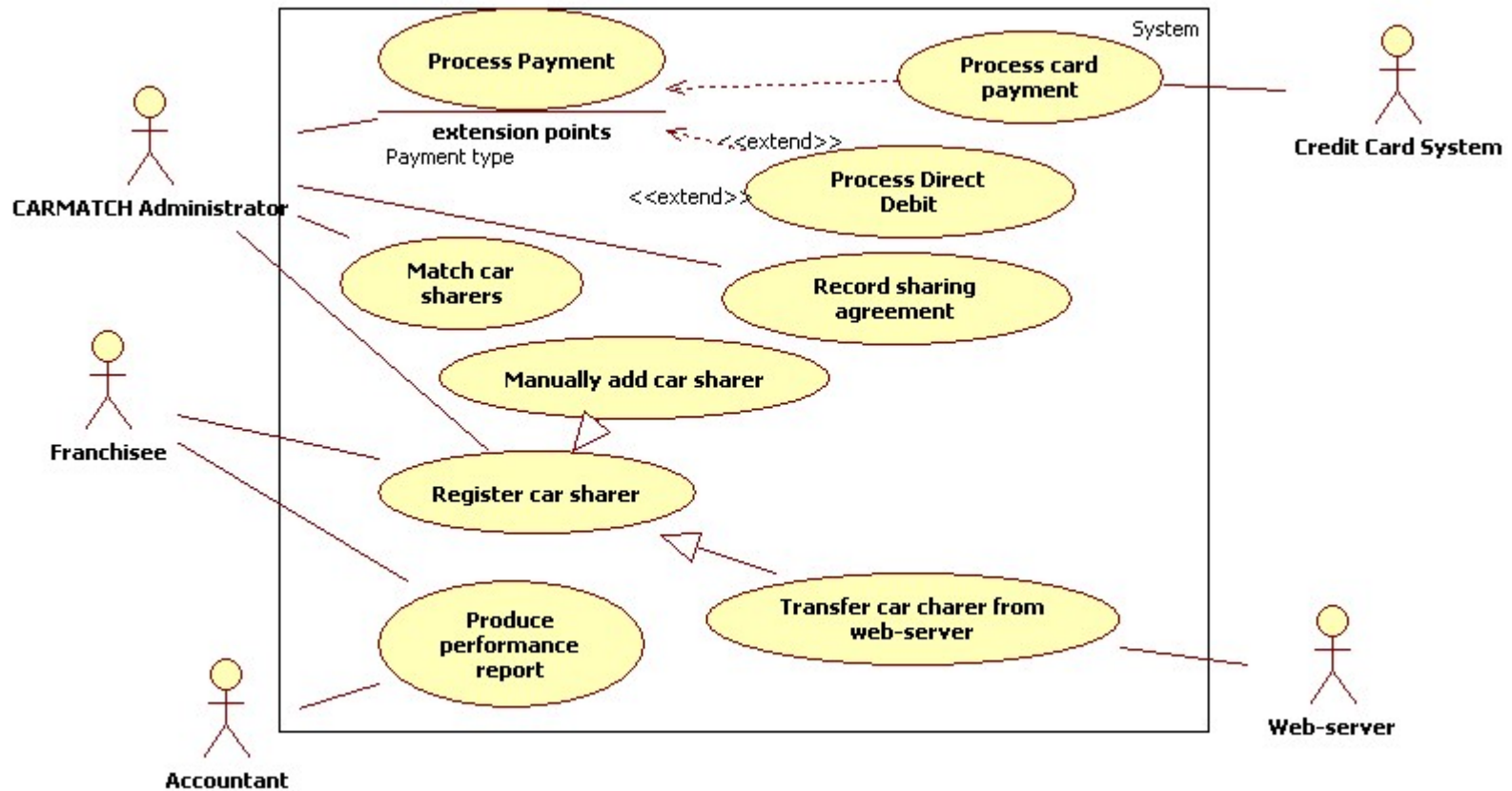
- **Project Drivers**
 - CARMATCH background
 - CARMATCH organization, Local governments, EU?, Local franchises, car sharers, etc.
 - System Users?
- **Project Constraints**
 - Budget, Deadlines, Laws, etc.?
- **Functional Requirements**
 1. To hold information about members
 1. 1
 2. 2
 3. 3
 2. To match car sharers
 3. To record insurance sales
 4. To record details consultancies
- **Non-functional Requirements**
- **Project Issues**



CARMATCH Actors and Use Cases



CARMATCH System Use cases



CARMATCH Use Case Description

Use Case: Register car sharer

Description: The registration of the car sharer information and the association with a membership number

Actors: CARMATCH Administrator, Car sharer

Assumptions: the CARMATCH Administrator has to confirm information, and the car sharer has to accept CARMATCH policy

Steps:

1. The CARMATCH Administrator enters name, address and all requested (mandatory) information of the car sharer in system entry window
2. The CARMATCH Administrator enters sharing information (e.g., time, starting address, car type, etc.)
3. ...

Variations (optional): any variations in the steps of a use case

Non-Functional (optional): List of non-functional requirements that the use case must meet.

Issues: List of issues that remain to be solved

Use Cases

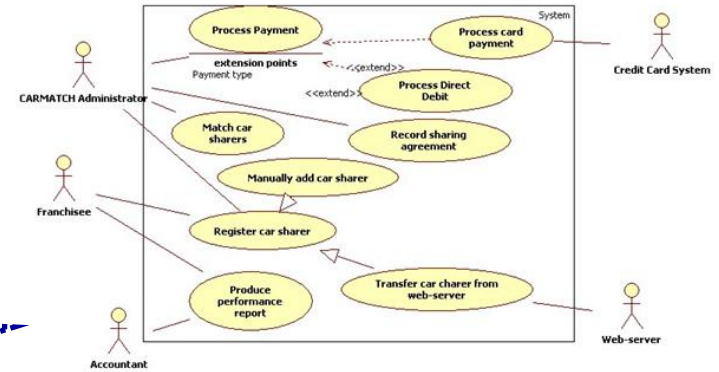
CARMATCH Requirements Specification

The System Requirements Specification

Table of Contents

- Project Drivers
 - CARMATCH background
 - CARMATCH organization, Local governments, EU, Local Franchisees, car sharers, etc.
 - System Users?
- Project Constraints
 - Budget, Deadlines, Laws, etc.?
- Functional Requirements
 - To hold information about members
 - 1.1
 - 2.2
 - 3.3
 - To match car sharers
 - To record insurance sales
 - To record details consultations
- Non-functional Requirements
- Project Issues

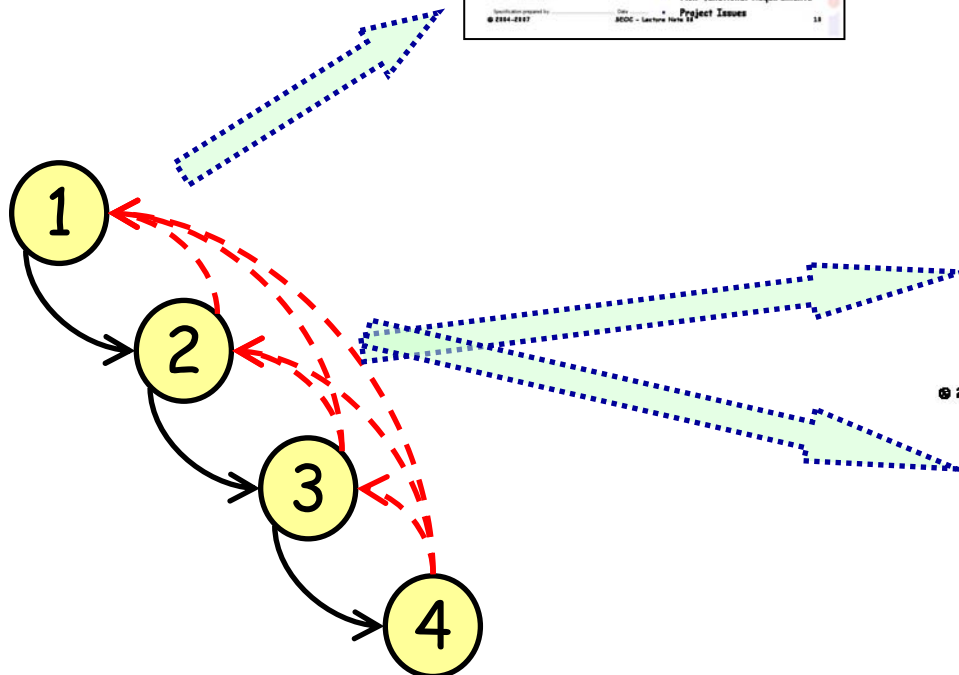
CARMATCH System Use cases



© 2004-2007

SEOC - Lecture Note 06

13



CARMATCH Use Case Description

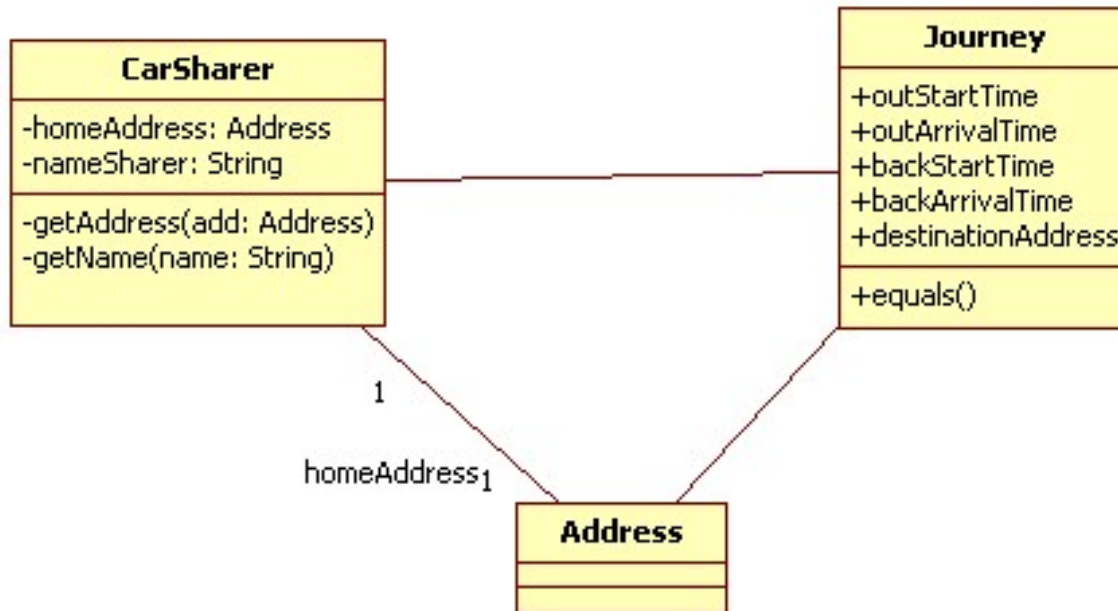
Use Case: Register car sharer
Description: The registration of the car sharer information and the association with a membership number
Actors: CARMATCH Administrator, Car sharer
Assumptions: the CARMATCH Administrator has to confirm information, and the car sharer has to accept CARMATCH policy
Steps: <ol style="list-style-type: none"> The CARMATCH Administrator enters name, address and all requested (mandatory) information of the car sharer in system entry window The CARMATCH Administrator enters sharing information (e.g., time, starting address, car type, etc.) ...
Variations (optional): any variations in the steps of a use case
Non-Functional (optional): List of non-functional requirements that the use case must meet.
Issues: List of issues that remain to be solved

© 2004-2007

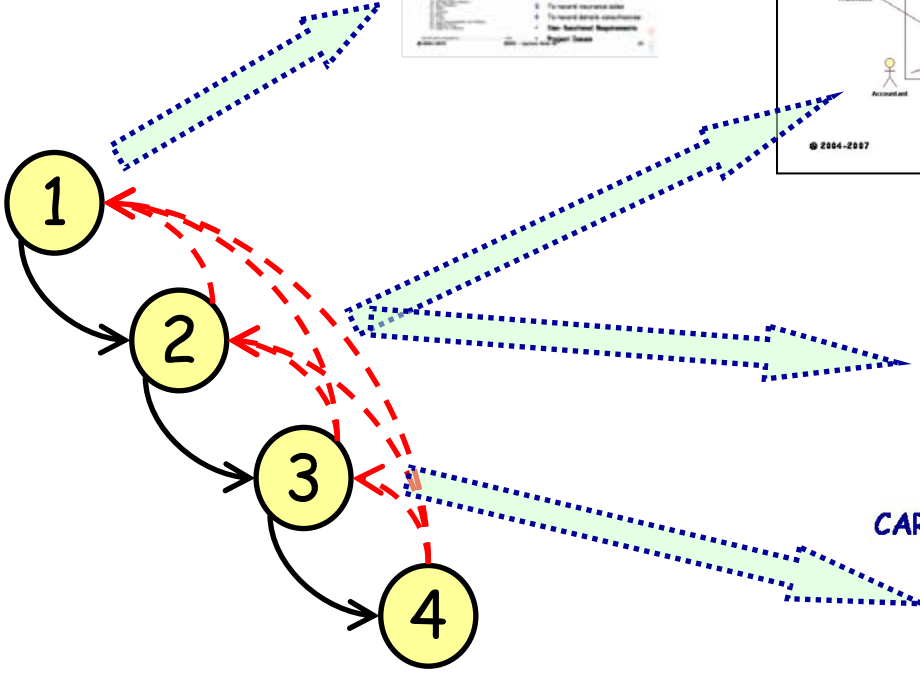
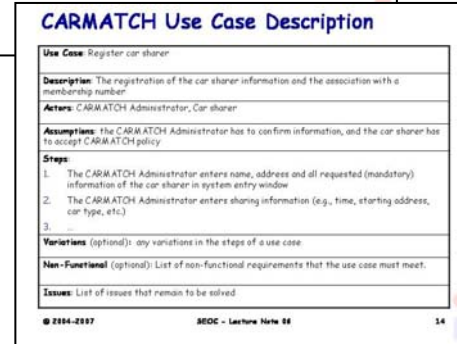
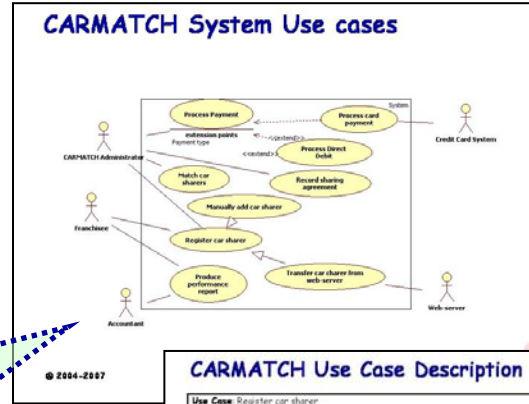
SEOC - Lecture Note 06

14

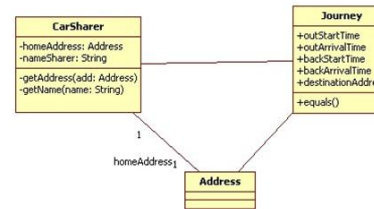
CARMATCH Class Diagram



Class Diagram



CARMATCH Class Diagram



CARMATCH Validation

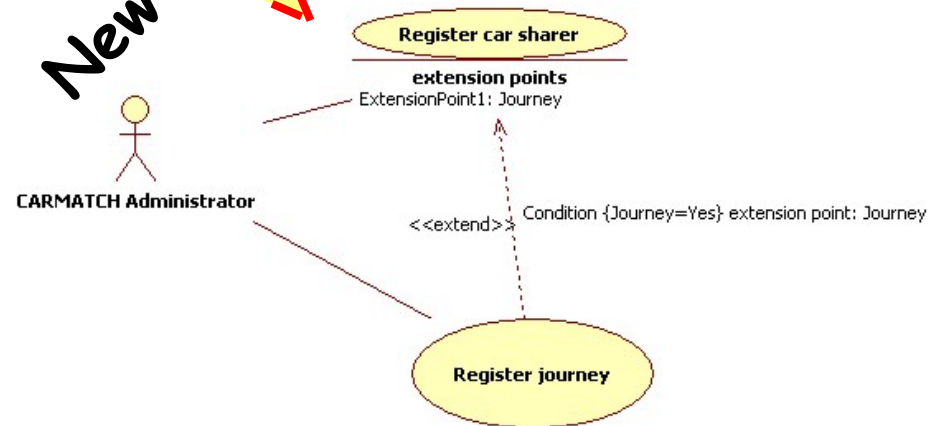
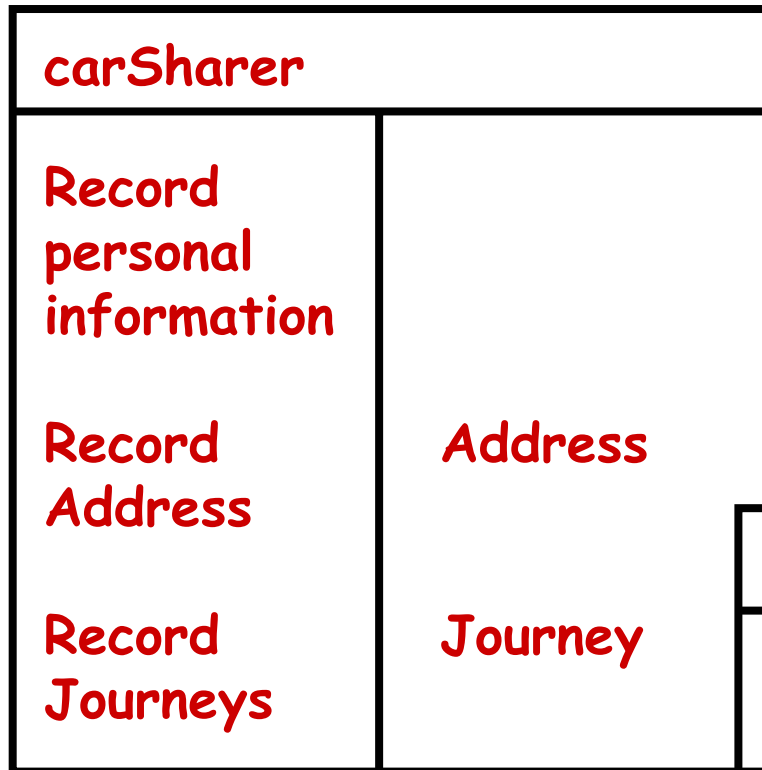
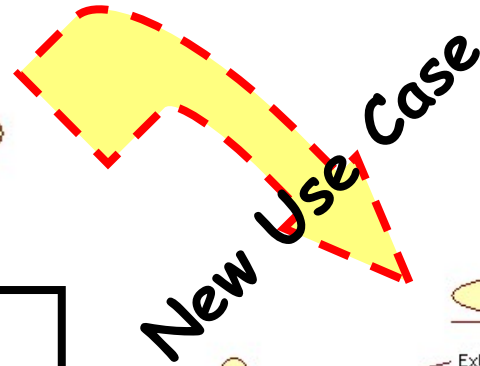
carSharer	
Record personal information	
Record Address	Address
Record Journeys	Journey

Journey	
Record journey Times	
Record journey Addresses	Address
Find journey matches	

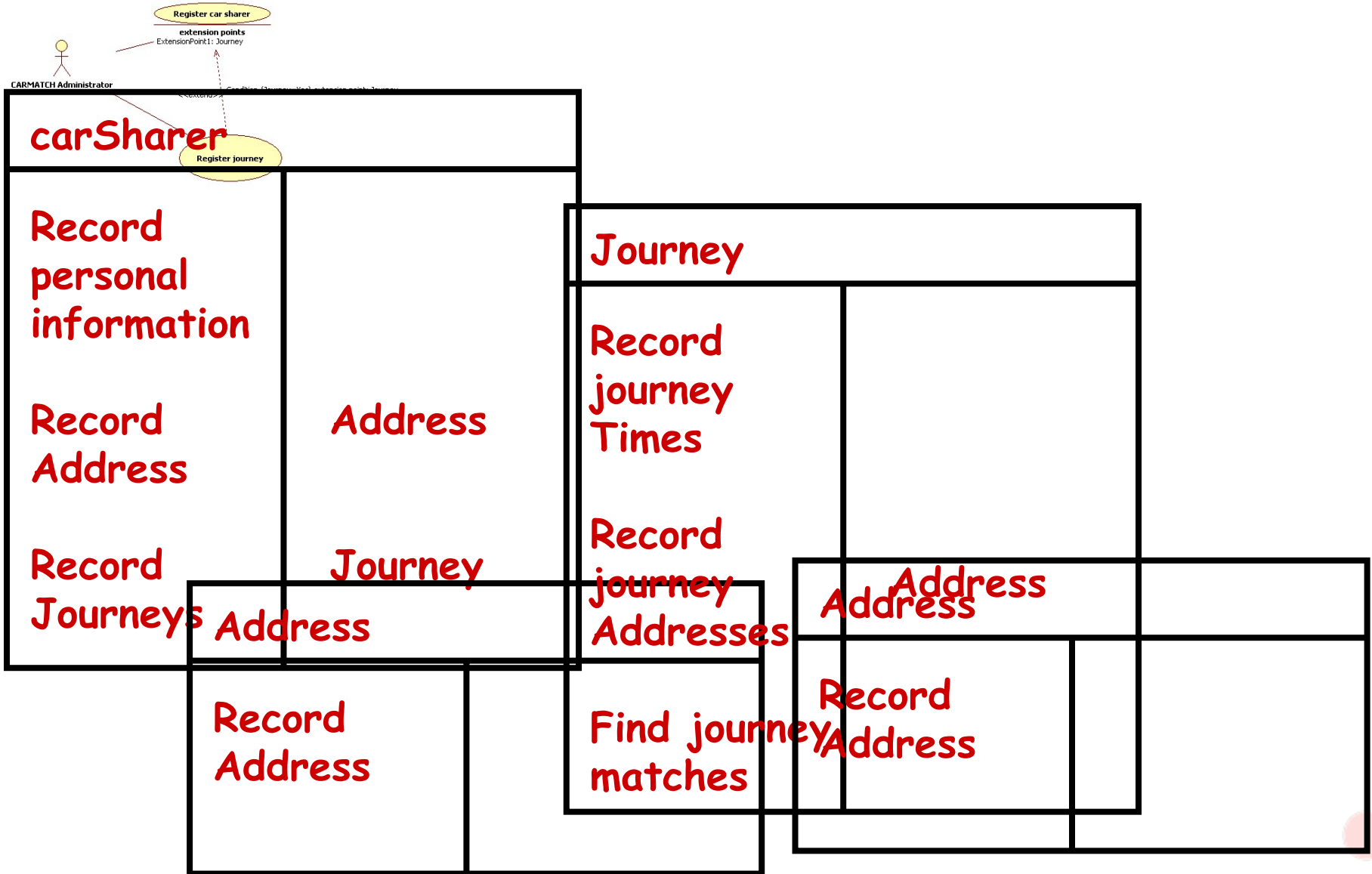
Address	
Record Address	



CARMATCH: Register car sharer



CARMATCH: Register Journey

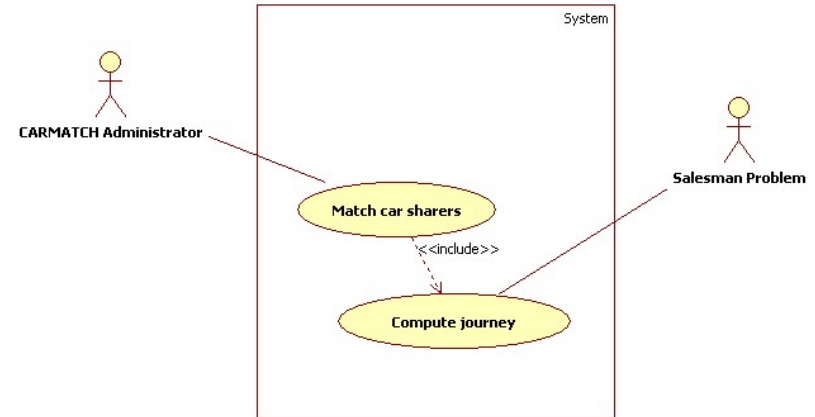
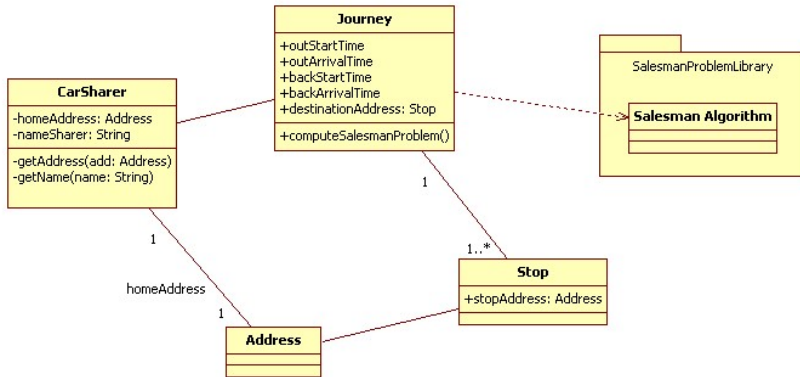


CARMATCH new requirements

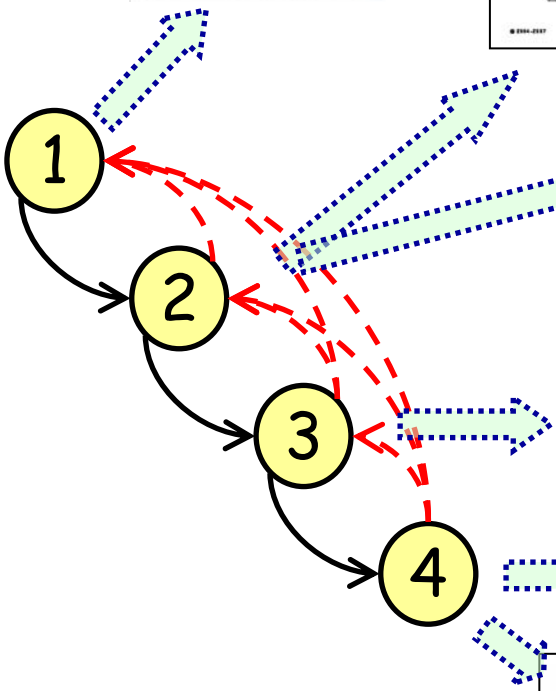
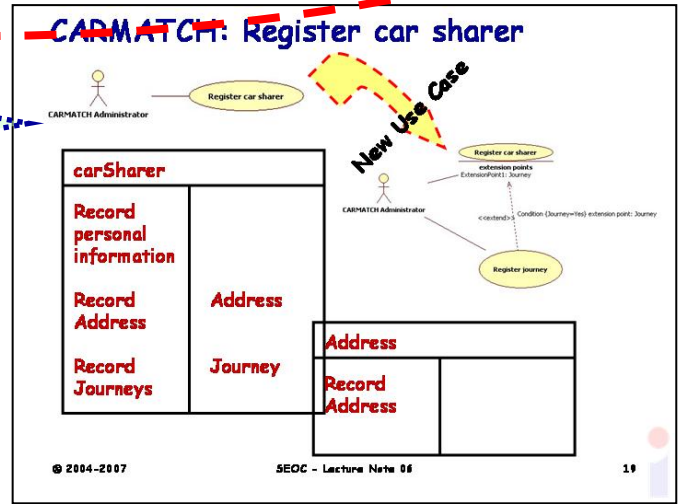
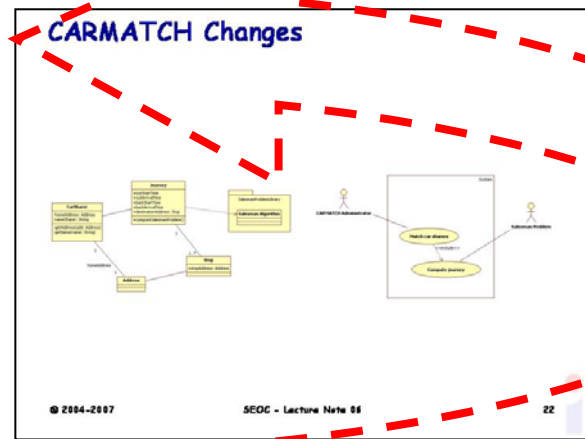
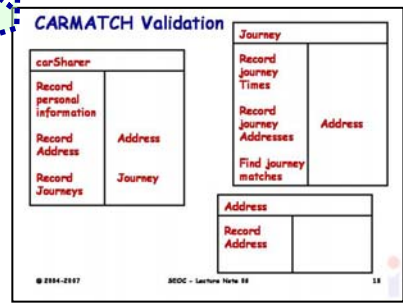
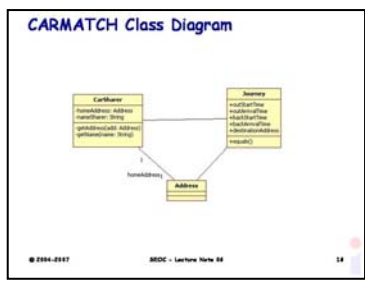
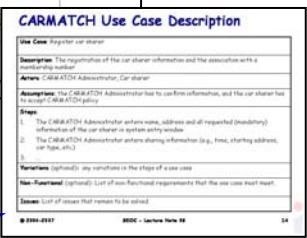
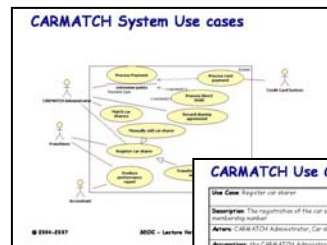
- Efficiency: maximize the combination of journeys by combining multiple stops (i.e., journeys)
- Note that it is a non-functional requirements
- Are there any implications? How does it affect your preliminary design?



CARMATCH Changes



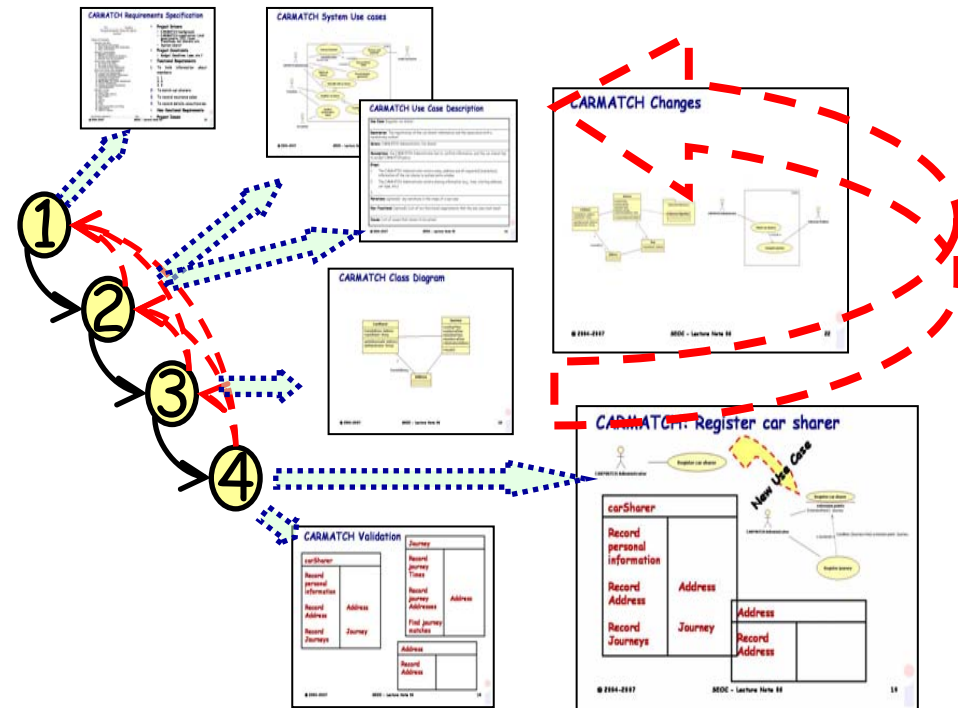
CARMATCH validation



Assessment

Part 3 - Deliverable Marking Scheme

Deliverable Marking Scheme		
Deliverable Part	Questions	Marks
Requirements Marks Limit: [20/100]	Q1. Did you organise/collect the system requirements by using a Requirements Specification template (e.g., Volere)? Assess the quality of your Software Requirements Specification (SRS) document.	[/ 5]
	Q2. Did you distinguish different types of requirements (e.g., functional or non-functional)? Assess how your SRS identifies different types of requirements.	[/ 5]
	Q3. Do you believe you got most of the system requirements (requirements completeness)? Assess the extent to which you have elicited and gathered requirements from the main sources.	[/ 5]
	Q4. Have you identified/resolved conflicting requirements (requirements correctness)? Assess the extent to which you have resolved conflicting requirements among different types (e.g., functional and non-functional) or across teams.	[/ 5]
Use Cases Marks Limit: [30/100]	Q5. Did you graphically represent the functional requirements by Use Cases? Assess to which extent your use case diagram captures main system functionalities and actors.	[/ 10]
	Q6. Did you refine the use cases by generalization, include or extend relationships? Assess to which extent you have refined and structured use cases.	[/ 10]
	Q7. Did you use a template for describing use cases? Assess to which extent you have clarified and described use case information (completeness and correctness).	[/ 10]
Class Diagrams Marks Limit: [30/100]	Q8. Does your class diagram identify the main classes of the system? Assess to which extent your class diagram realizes system use cases.	[/ 10]
	Q9. Did you specify Attributes and Operations for each class? Assess the completeness of class specification.	[/ 10]
	Q10. Did you identify relationships (i.e., Dependency, Association, Aggregation, Composition and Inheritance or Generalization) between classes? Assess the object orientation quality of your class diagram.	[/ 10]
CRC Cards Marks Limit: [20/100]	Q11. Did you construct CRC cards for your system design? Assess the completeness and correctness of CRC cards.	[/ 10]
	Q12. Did you verify your Class Diagrams? Did you play any use case with the CRC Cards in order to verify your class diagram? Assess the quality and the coverage of your requirements and design verification by CRC cards.	[/ 10]
Deliverable Mark		[/100]



Summary

- The SEOC process so far
 - Requirements, Use Cases, Class Diagram, Validation
 - Iterations, Feedbacks
 - Assessment
- CARMATCH: Sample Designs
- Assessment

