Visual Paradigm for UML

AN INTRODUCTORY TUTORIAL

Mario G.C.A. Cimino

Topics

1. Textual Analysis

2. Class Diagram (UML)

3. Use Case Diagram (UML)

4. Use Case Details (Enterprise Edition)

5. Sequence Diagram (UML)

6. User interface Diagram (Enterprise Edition)

7. Brainstorm Diagram

8. Report generation

9. Support to Model-Driven approach (MDA).

POSSIBLE WRITTEN QUESTIONS
FOR EXAMS.
WHAT IS:
* TEXTUAL ANALYSIS
* CLASS DIAGRAM
* USE CASE DIAGRAM
* USE CASE DETAILS
* SEQUENCE DIAGRAM
* USER INTERFACE DIAGRAM
* BRAINSTORM DIAGRAM

PROVIDE SIMPLE ANSWERS, INSPIRED
BY THIS TUTORIAL

1. Textual Analysis

1. Diagram Navigator \ Requirements Capturing \
2. Right click on Textual Analysis → *New Textual Analysis*
3. Paste or import textual description
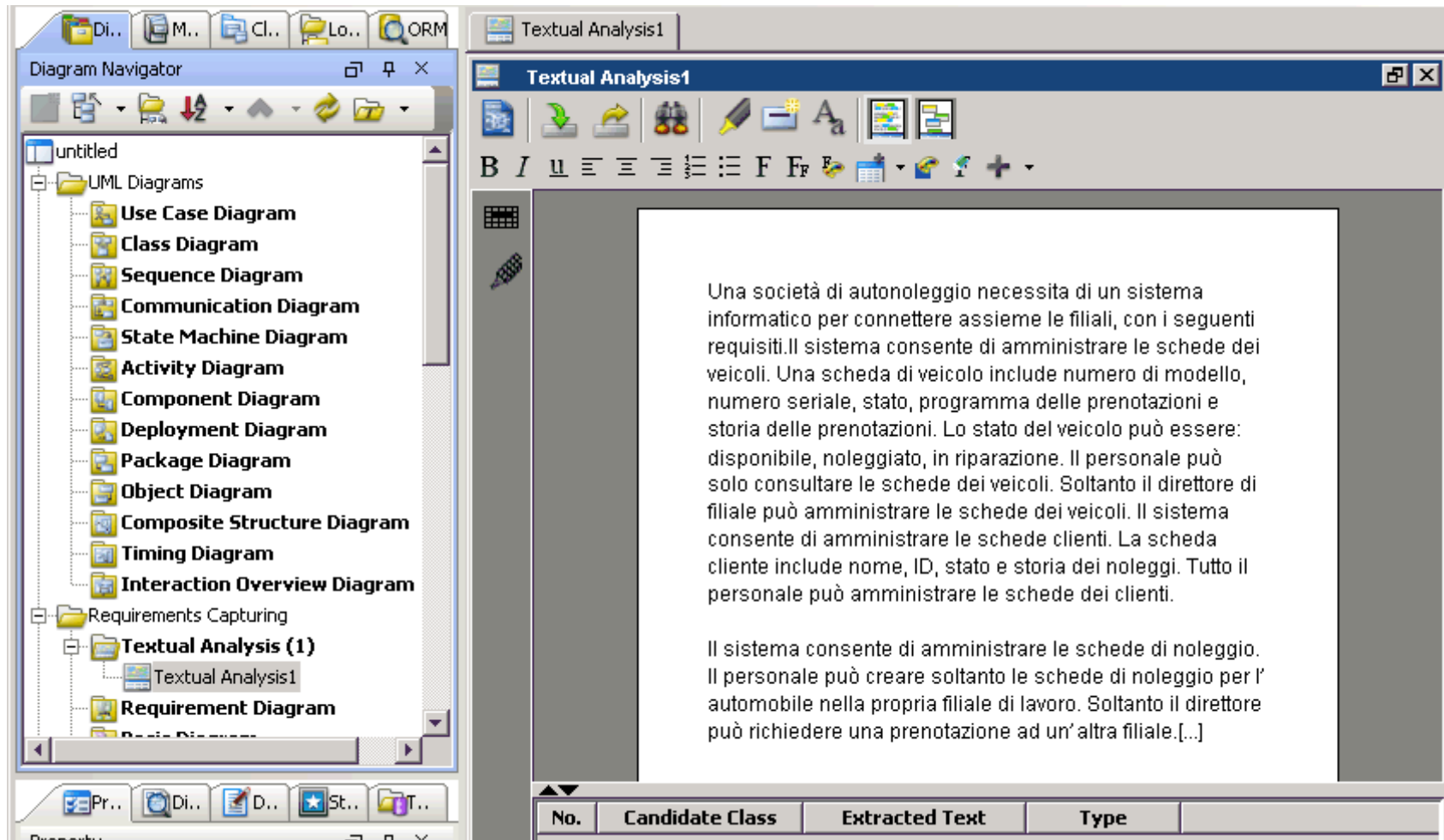
4. Right click on the term *veicolo/veicoli* → Class
5. Right click on the term *cliente/clienti* → Class
6. In the table below the text, make uniform the field *candidate class* for singular/plural forms
(*camel case format*)
7. Right click on the text *storia delle prenotazioni* → Class
8. Right click on the term *Personale* → Actor *Impiegato*
9. Right click on the text *amministrare le schede dei veicoli* → Use case *AmministraSchedeVeicoli*

Una società di autonoleggio necessita di un sistema informatico per connettere assieme le filiali, con i seguenti requisiti.Il sistema consente di amministrare le schede dei veicoli. Una scheda di veicolo include numero di modello, numero seriale, stato, programma delle prenotazioni e storia delle prenotazioni. Lo stato del veicolo può essere: disponibile, noleggiato, in riparazione. Il personale può solo consultare le schede dei veicoli. Soltanto il direttore di filiale può amministrare le schede dei veicoli. Il sistema consente di amministrare le schede clienti. La scheda cliente include nome, ID, stato e storia dei noleggi. Tutto il personale può amministrare le schede dei clienti.
Il sistema consente di amministrare le schede di noleggio. Il personale può creare soltanto le schede di noleggio per l'automobile nella propria filiale di lavoro. Soltanto il direttore può richiedere una prenotazione ad un'altra filiale.[...]

No.	Candidate Class	Extracted Text	Type
1	Veicolo	veicoli	Class
2	Veicolo	veicolo	Class
3	Cliente	cliente	Class
4	Cliente	clienti	Class
5	StoriaPrenotazioni	storia delle prenotazioni	Class
6	Impiegato	personale	Class
7	AmministraSchedeVeicoli	amministrare le schede dei veicoli	Use Case

2. Class Diagram

1. Right click on the row corresponding to a candidate element → *Create Class Model Element* → Create ClassDiagram1
2. If a model element with the same name already exists, then the system asks whether the existing model element can be employed (e.g. for synonym, plural/singular forms)

3. A model element which does not appear in any diagram is called *hidden* element. A classifier which appears in a diagram without a corresponding model element is called *ghost element*.
4. Generated elements appear in the *Model Explorer* (on the left).

Model Explorer

untitled
- AmministraSchedeVeicoli
- Candidate Items
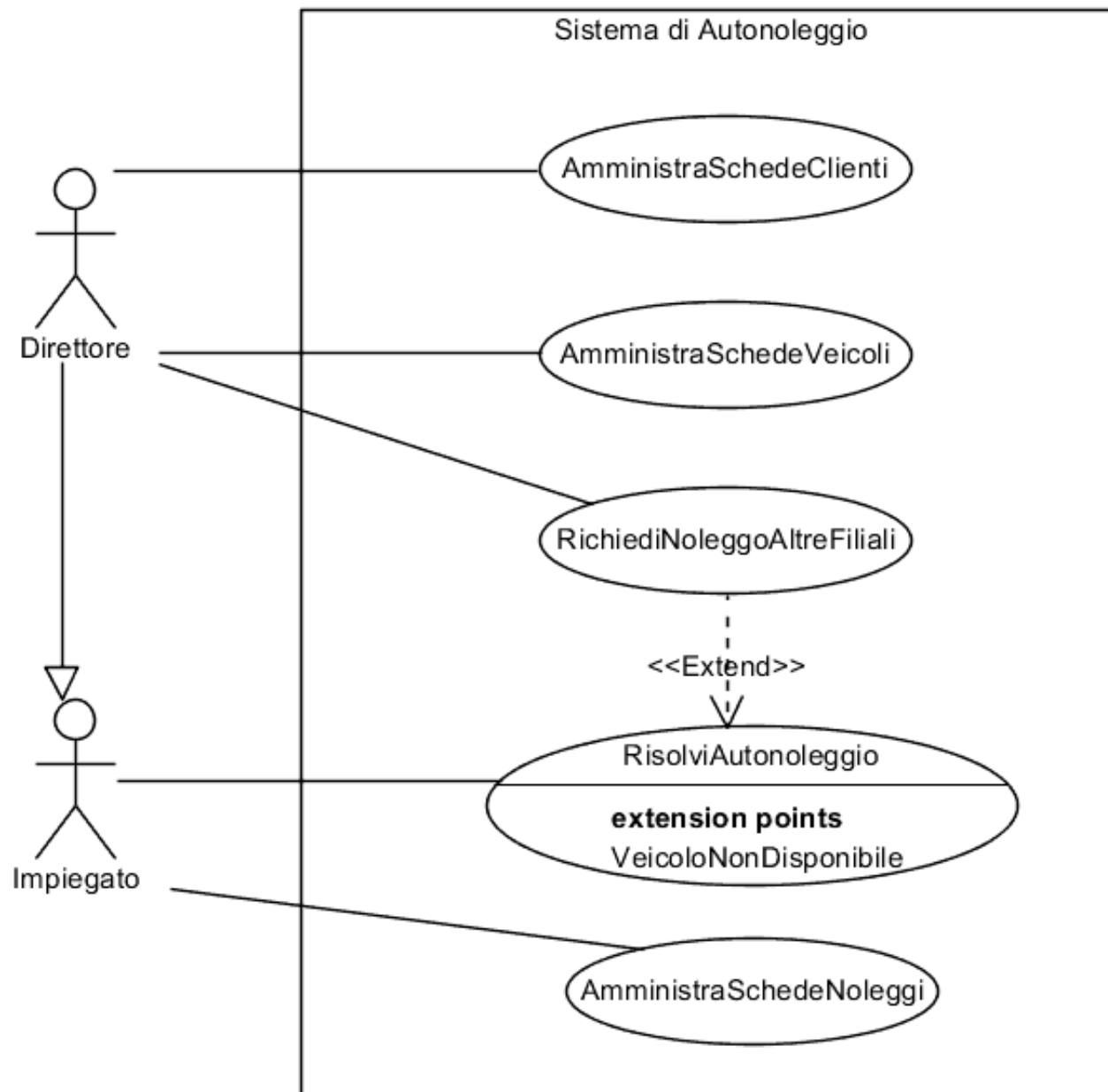- Cliente
- Impiegato
- Impiegato
- StoriaPrenotazioni
- Veicolo

stato del veicolo può essere: disponibile, noleggiato, in riparazione. Il personale può solo consultare le schede dei veicoli. Soltanto il direttore di filiale può amministrare le schede dei veicoli. Il sistema consente di amministrare le schede clienti. La scheda cliente include nome, ID, stato e storia dei noleggi. Tutto il personale può amministrare le schede dei clienti.
Il sistema consente di amministrare le schede di noleggio. Il personale può creare soltanto le schede di noleggio per l'automobile nella propria filiale di lavoro. Soltanto il direttore può richiedere una prenotazione

No.	Candidate Class	Extracted Text	Type
1	Veicolo	veicoli	Generated Model Element
2	Veicolo	veicolo	Generated Model Element
3	Cliente	cliente	Generated Model Element
4	Cliente	clienti	Generated Model Element
5	StoriaPrenotazioni	storia delle prenotazioni	Generated Model Element
6	AmministraSchedeVeicoli	amministrare le schede dei veicoli	Generated Model Element
7	Impiegato	personale	Generated Model Element

5. Model elements can be dragged to diagrams, generating views.
6. Drag the *Veicolo* class to *ClassDiagram1* diagram.
7. Right click on the class → Add Attribute.
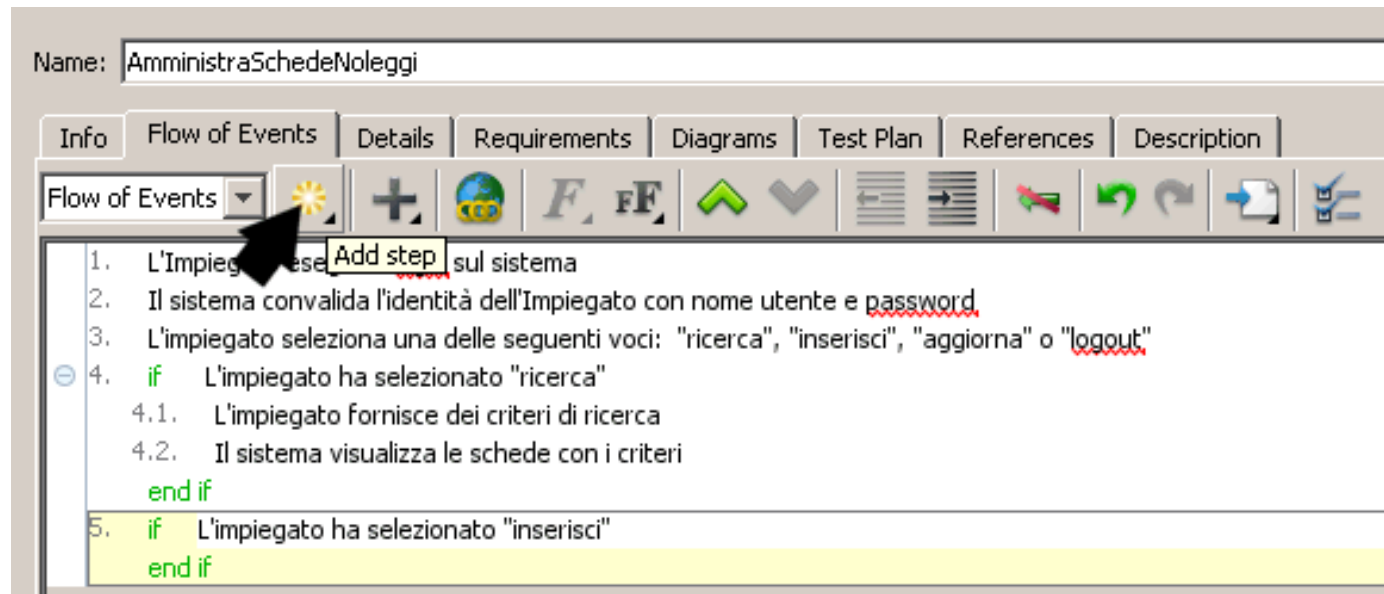8. Add the attributes of a vehicle.

3. Use Case Diagram

1. Right click on the Diagram Navigator \ Use case Diagram → *New Use Case Diagram.*
2. Drag model elements from the Model Explorer.
3. Create new elements in the diagram using the toolkit (on the left).
4. To create a relationship between elements: hover mouse over an element, choose the relationship, drag the chosen relationship to the other element.
5. Define Direttore as an extension of Impiegato (a director is as an employee with other additional capabilities. A director can take the place of any employee (*substitutability principle* in object-oriented models). As an effect, redundancy in the diagram is reduced.
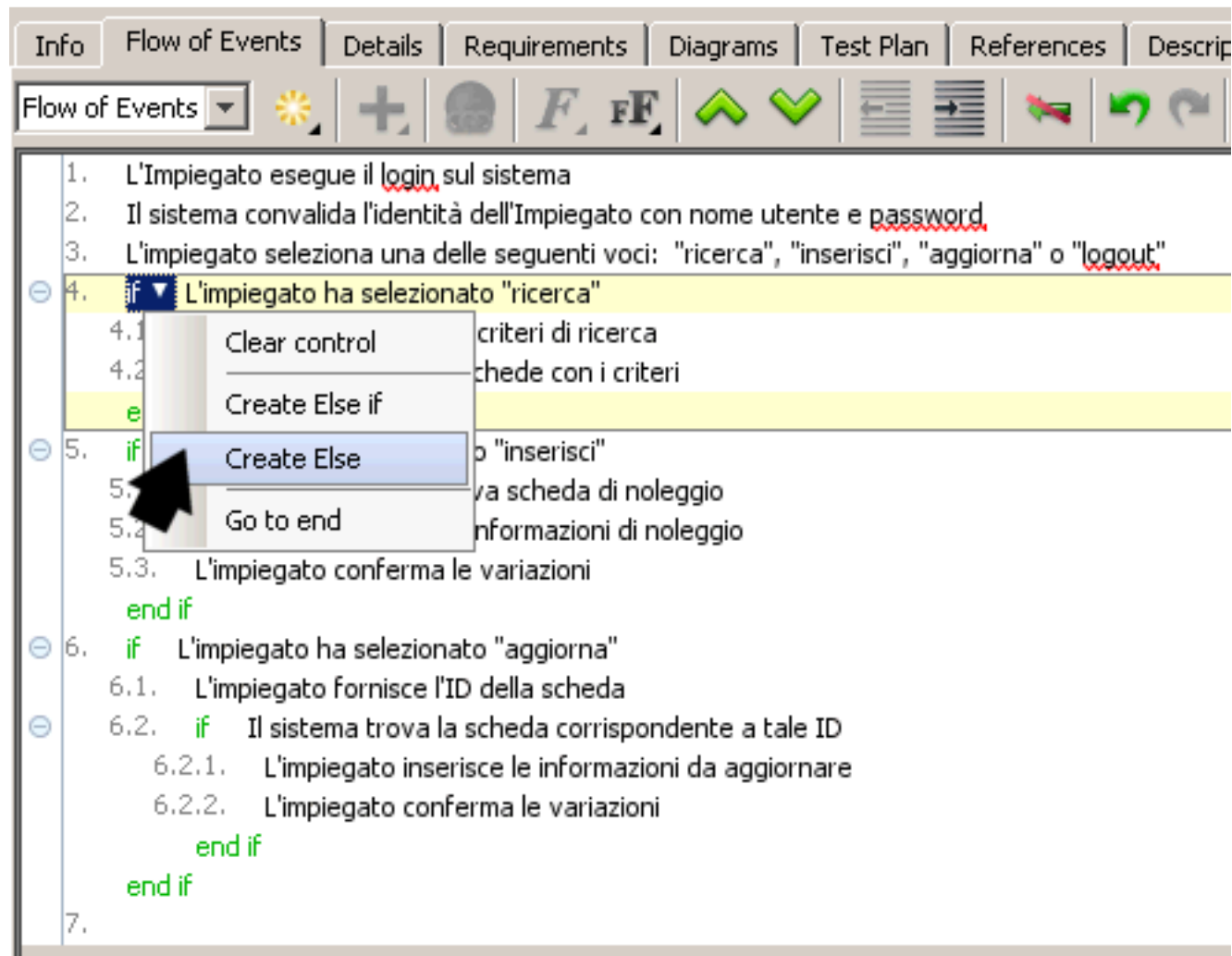6. Define an extension point over the *RisolviAutonoleggio* use case.

Sistema di Autonoleggio

Direttore

AmministraSchedeClienti

AmministraSchedeVeicoli

RichiediNoleggoAltreFiliali

<<Extend>>

RisolviAutonoleggio

extension points
VeicoloNonDisponibile

Impiegato

AmministraSchedeNoleggi

4. Use Case Details

1. Right click on the AmministraSchedeNoleggi use case → *Open UseCase Details…*
2. Tab Info → Primary Actors: *Impiegato*
3. Tab Info → Documentation: *L'impiegato amministra le schede di noleggio*
4. Tab Details → Preconditions: *L'impiegato è connesso al sistema*
5. Tab Details → Postconditions: *L'impiegato non è connesso al sistema*
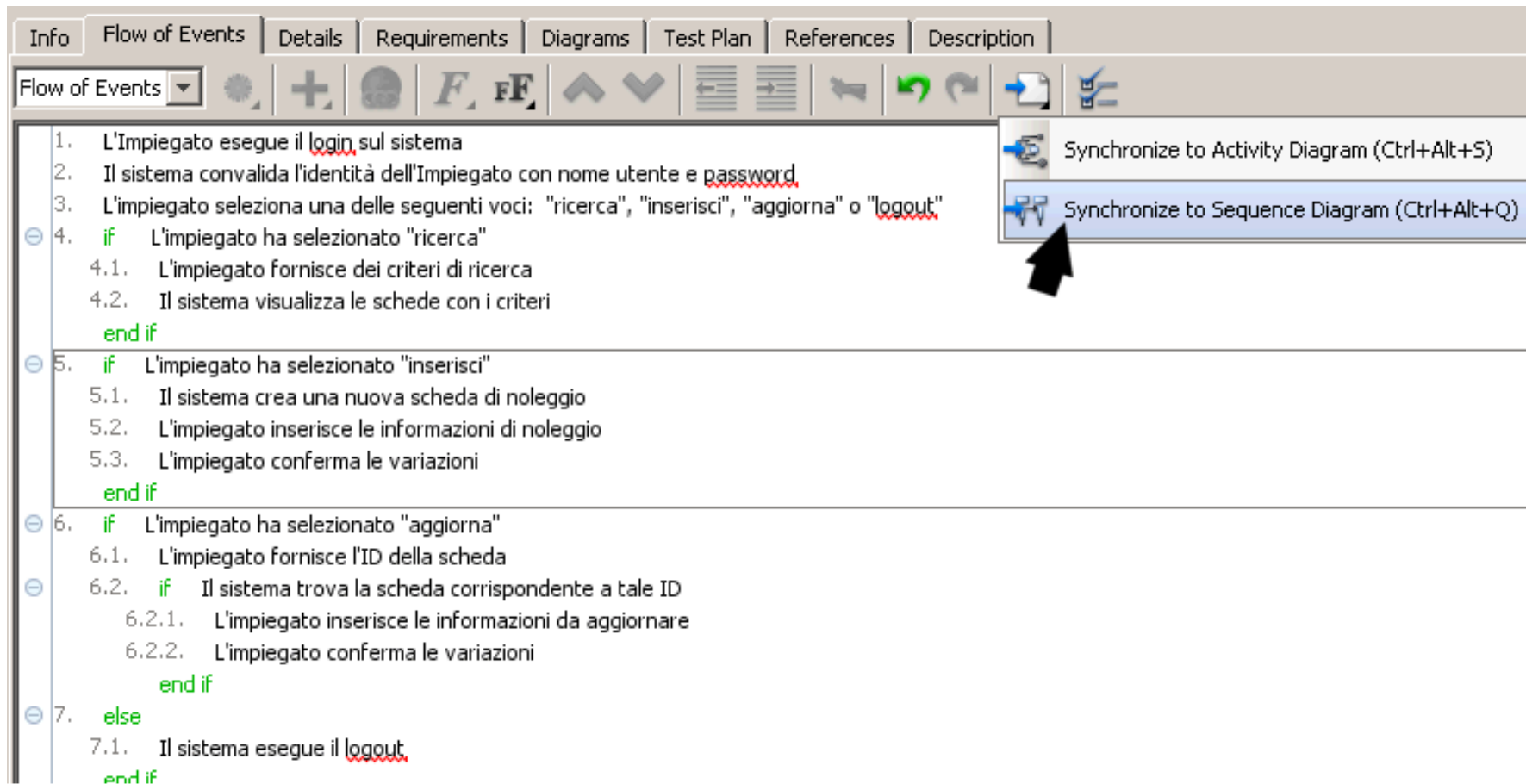6. Tab *Flow of events* → Insert the main flow. For alternative flows use the *add step* button:

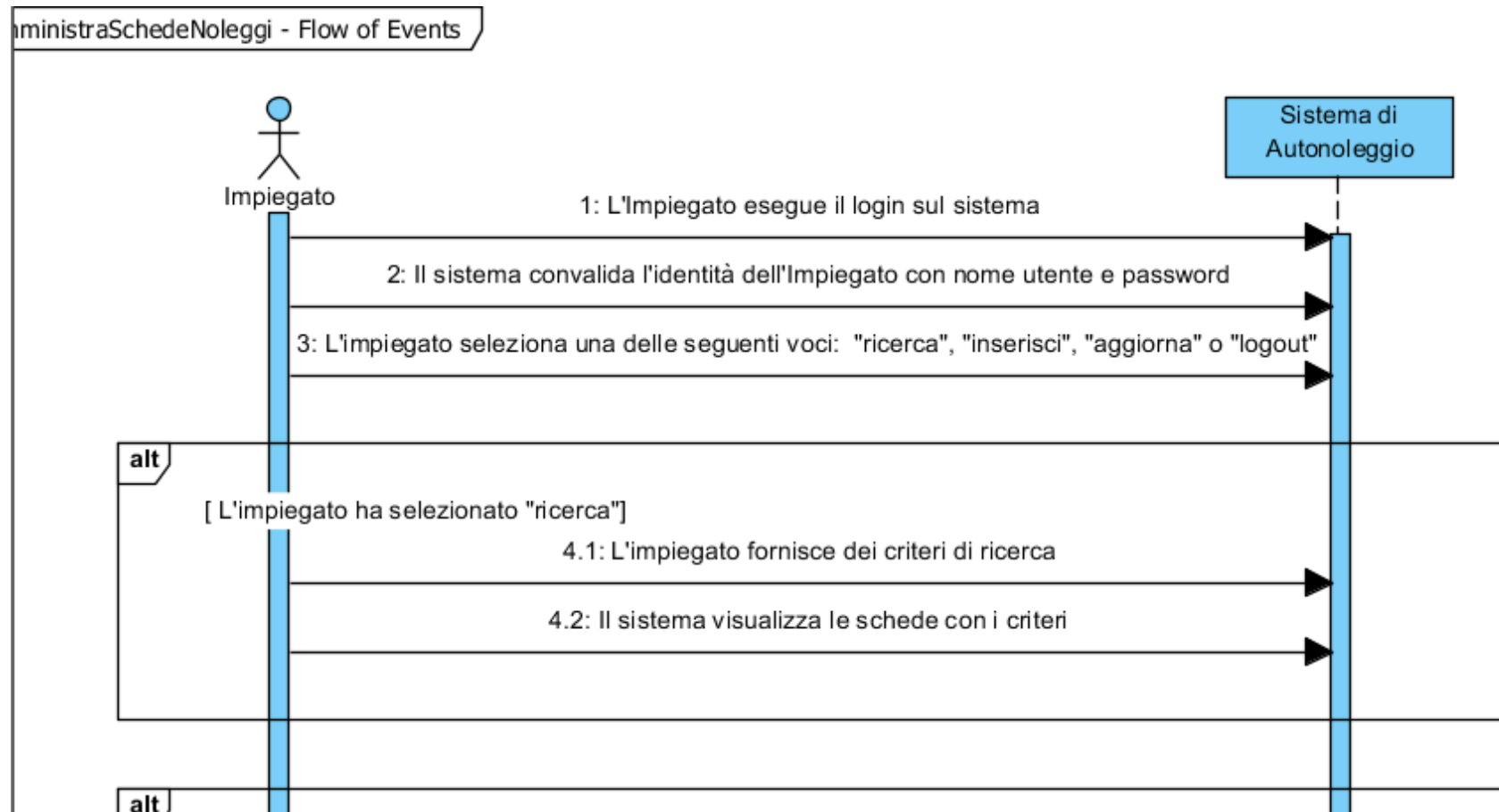*7.* To create an *Else* branch, hover mouse over an *If* branch and select *Create Else*.

Caso d'uso	AmministraSchedeNoleggi
Use Case ID	UC-100
Super Use Case	Nessuno
Brief Description	L'impiegato amministra le schede di noleggio
Primary Actor	Impiegato
Secondary Actor(s)	Nesuno
Preconditions	L'impiegato è connesso al sistema

Main flow	1. L'Impiegato esegue il login sul sistema
	2. Il sistema convalida l'identità dell'Impiegato con nome utente e password
	3. L'impiegato seleziona una delle seguenti voci: "ricerca", "inserisci", "aggiorna" o "logout"
	4. IF L'impiegato ha selezionato "ricerca"
	    4.1 L'impiegato fornisce dei criteri di ricerca
	    4.2 Il sistema visualizza le schede con i criteri
	5.IF L'impiegato ha selezionato "inserisci"
	    5.1 Il sistema crea una nuova scheda di noleggio
	    5.2 L'impiegato inserisce le informazioni di noleggio
	    5.3 L'impiegato conferma le variazioni
	6.IF L'impiegato ha selezionato "aggiorna"
	    6.1 L'impiegato fornisce l'ID della scheda
	    6.2 IF Il sistema trova la scheda corrispondente a tale ID
	        6.2.1 L'impiegato inserisce le informazioni da aggiornare
	        6.2.2 L'impiegato conferma le variazioni
	7.ELSE
	    7.1 Il sistema esegue il logout

Postconditions	L'impiegato non è connesso al sistema
Alternative flows	Nessuno
Author	Alan Turing
Date	13/10/2005 15:30

5. Sequence Diagram
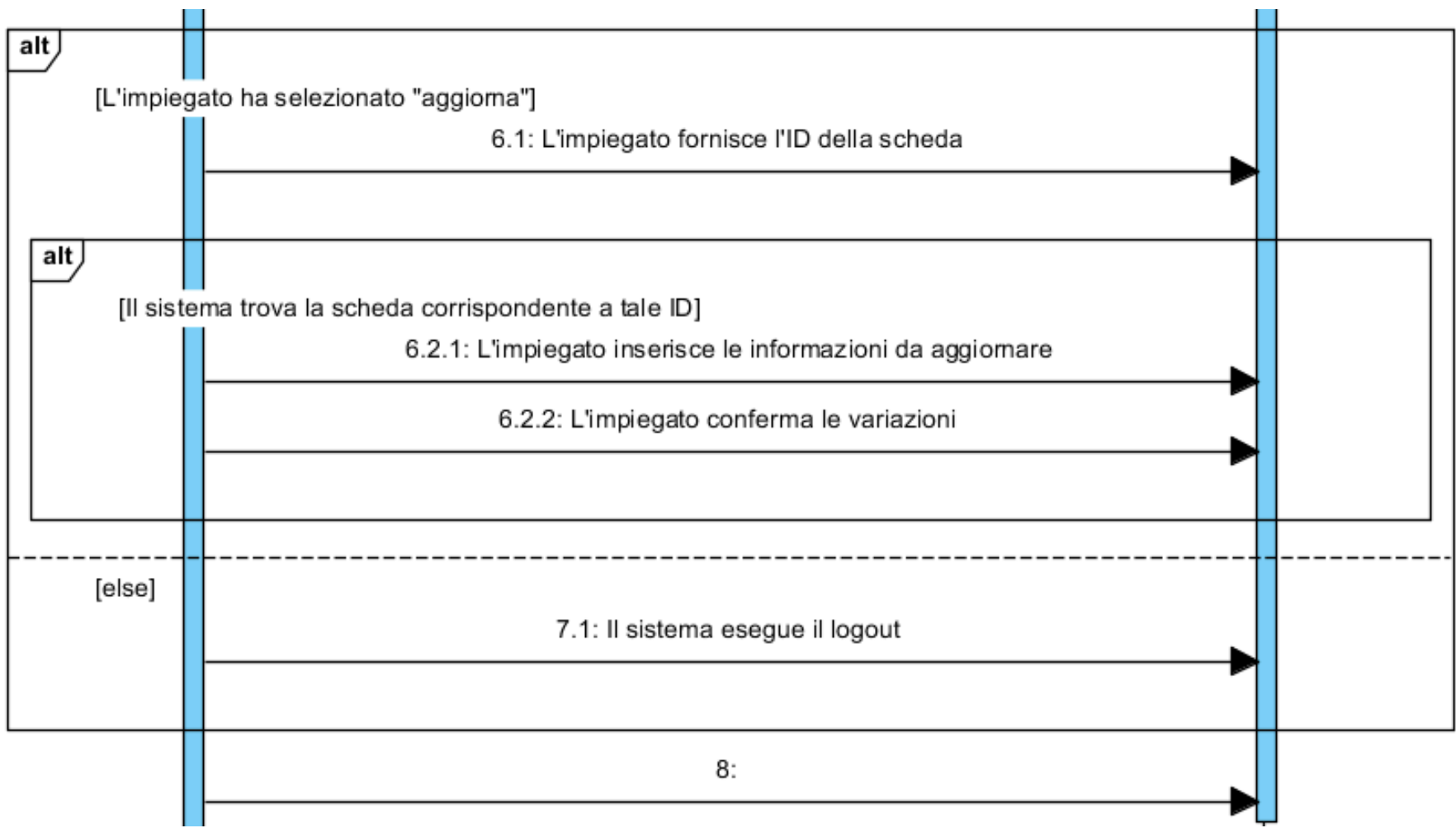
1. Create a sequence diagram related to the *AmministraSchedeNoleggi* use case.
2. Tab *Flow of events* → button *Synchronize to Sequence Diagram*

3. The sequence diagram is created and shown.

...

alt

[L'impiegato ha selezionato "aggiorna"]

6.1: L'impiegato fornisce l'ID della scheda

alt

[Il sistema trova la scheda corrispondente a tale ID]

6.2.1: L'impiegato inserisce le informazioni da aggiornare

6.2.2: L'impiegato conferma le variazioni

[else]

7.1: Il sistema esegue il logout

8:

6. User Interface Diagram

1. Create a screen mockup related to the *Veicolo* record.
2. Right button on the *Veicolo* class → Subdiagrams → Others → User interface
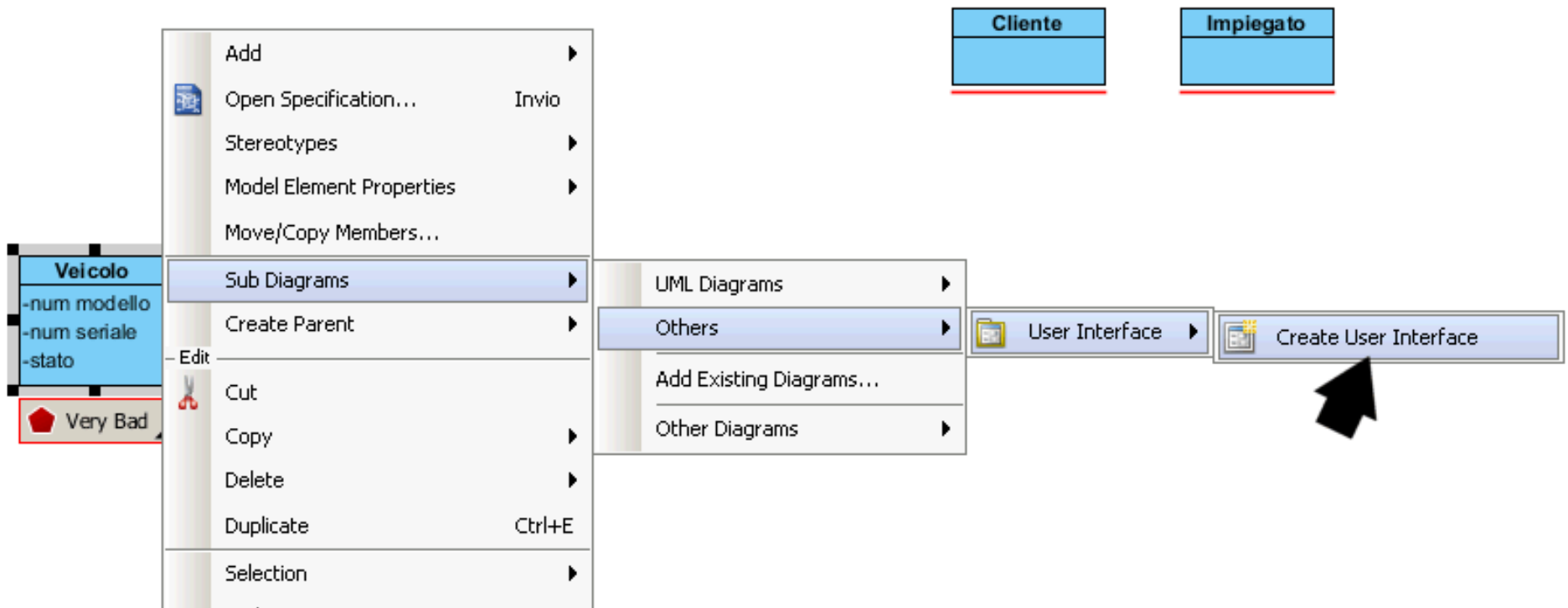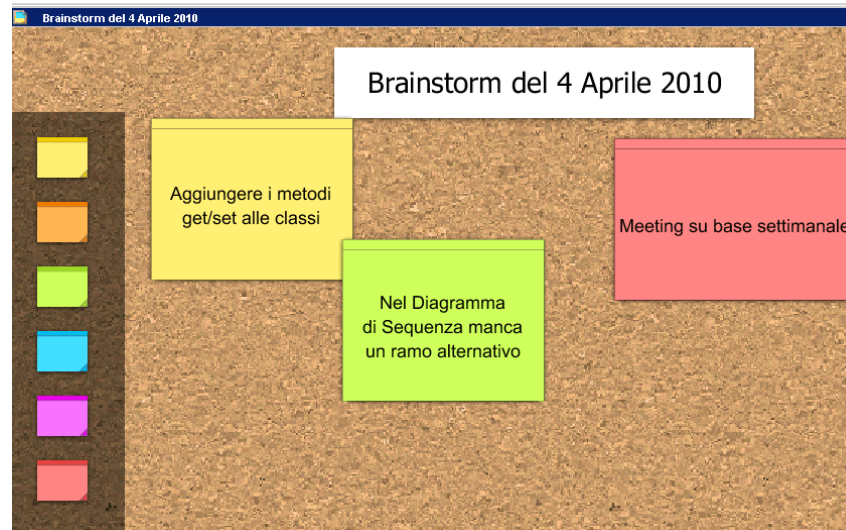

3. Use the toolkit to create an interface

7. Brainstorm Diagram

1. Allows managing a panel via post-it, to be connected to diagrams.

8. Managing reports

1. Tools → Report → Generate HTML Report

9. Support to Model-Driven approach (MDA)

2. Generate code for specific platforms from a model, and viceversa.
3. Menu Tools → Code Engineering → Instant Generator → Java → Selezionare la classe → Preview.
4. The attributes of the *Veicolo* class must be expressed in camel case format