



UNIVERSITÀ DEGLI STUDI DI MILANO  
DIPARTIMENTO DI INFORMATICA

## Corso di Progettazione model-driven del software (Ing SW - Mod 2)

*Laboratorio 1 - Informazioni generali sul  
laboratorio e esempi di Use Case Diagram*

`chiara.braghin@unimi.it`

1

## **Introduzione al laboratorio**

2

## Organizzazione dei laboratori (1)

**24 ore in 6 settimane => 4 ore settimanali**

- 2 hr su **Zoom** (solo io): ripasso, "pillole" di teoria e esercizio caso di studio svolto insieme
  - **Lezione SINCRONA, REGISTRATA (ricordatemelo!!)**
  - **ORARIO:** giovedì, ore 11:30-13:30 (si inizia puntuali)
    - Registrazioni disponibili su Ariel
  - Link: <https://zoom.us/j/94255671991?pwd=TkFzSmphNG5uSXdkZ5Y1RnNDFOU09>
- 2 hr su **Discord** (con i tutor): esercizi che gli studenti svolgono individualmente o in gruppo
  - **Lezione SINCRONA, NON REGISTRATA**
  - **ORARIO:** giovedì, ore 14:30-16:30 (si inizia puntuali)

3

## Organizzazione dei laboratori (2)

Settimana	Argomenti
<b>WEEK 1 – 9-13/11/20</b>	<b>Diagrammi dei casi d'uso per l'analisi dei requisiti (scenari e storie), specifica dei requisiti</b>
<b>WEEK 2 – 16-20/11/20</b>	Diagrammi di sequenza vs diagrammi di attività
<b>WEEK 3 – 23-27/11/20</b>	Macchine di stato vs diagrammi attività
<b>WEEK 4 – 30/11- 4/12/20</b>	Java e persistenza dei dati (JDBC, Hibernate), Java Swing
<b>WEEK 5 – 7-11/12/20</b>	Design pattern MVC e DAO
<b>WEEK 6 – 14-18/12/20</b>	Altri design pattern Diagramma delle componenti e di deployment Dai modelli al codice

4

## Organizzazione dei laboratori (3)

### Maggiori informazioni su lezione pomeridiana:

- Gli esercizi da svolgere verranno postati su Ariel (ambiente *Laboratorio Mod 2 / Progettazione Model-Driven del SW (AA 2020/2021)*)
  - Fortemente consigliato il lavoro di gruppo!!!
- Non sempre prevista una connessione video con tutta la classe:
  - come nei laboratori in presenza potere arrivare "in classe" e iniziare a lavorare sugli esercizi proposti per quel giorno
- Saranno con me **due tutor**:
  - Mario Lilli, studente di dottorato
  - Davide Santandrea, studente della LM in Sicurezza Informatica

5

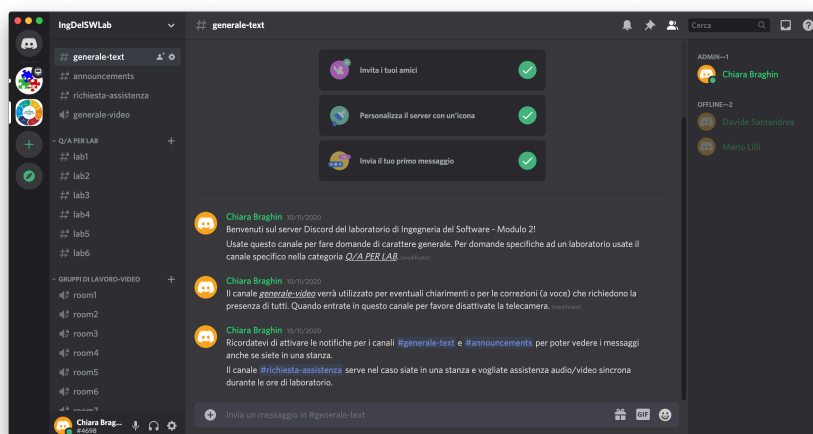
## Organizzazione dei laboratori (4)

### Come avere un feedback sul vostro lavoro?

- **Durante la lezione:** chiederci di venire nella vostra room (avremo audio, video e screen sharing disponibile)
- **Al di fuori della lezione:**
  1. Scrivere sui vari canali le vostre domande
  2. Avviare una chat con me o con i tutor
  3. Caricare le vostre soluzioni sul sito <https://upload.di.unimi.it/>
    - **SESSIONE:** Ing. SW -- Progettazione Model-driven del SW (Esercizi proposti durante i laboratori)
  4. Se vi serve una risposta sincrona, chiedere un appuntamento via chat (**NO MAIL**)

6

## Uso di Discord



7

## Strumenti/Applicazioni necessari

- **StarUML**
  - <https://staruml.io/>
  - Open source software modeling tool that supports UML (Unified Modeling Language)
- **DISCORD**
  - <https://discord.com/>
  - Piattaforma più flessibile e democratica per la gestione dei lavori di gruppo
- **GitHub**
  - <https://github.com/>

8

## Da fare per la lezione del pomeriggio

- Installare **StarUML**
- Fare account e installare **Discord**
  - Link di invito per accedere al server *IngDelSWLab*:  
<https://discord.gg/Y6Wcm57vwv>

9

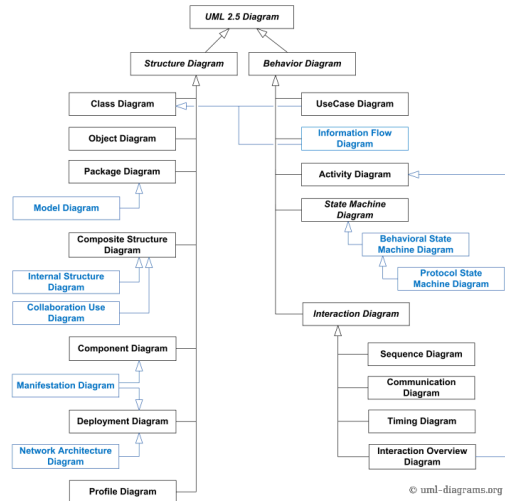
## LAB 1

### DIAGRAMMI DEI CASI D'USO

10

## Ripasso (1)

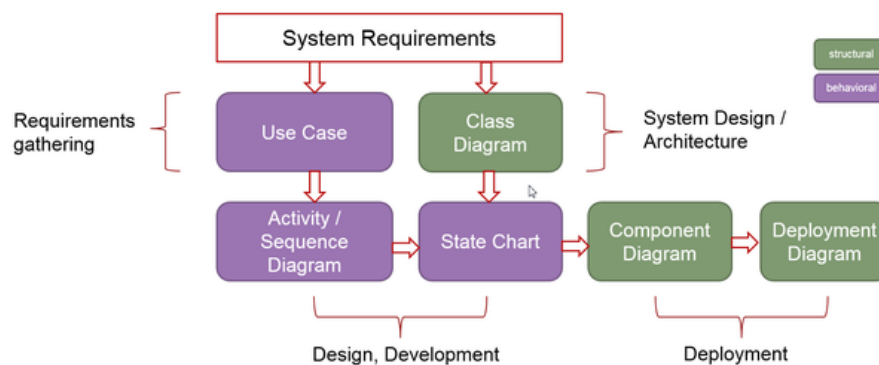
### UML a supporto di uno sviluppo del SW model-driven



11

## Ripasso (2)

### Models: Which and when?



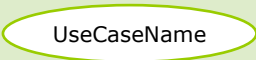
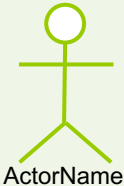
12

## Use Case Diagram – Casi d'uso

- Punto di partenza della progettazione del sistema
- Chiarisce (anche con il committente) cosa deve fare il sistema
  - **Requisiti funzionali** espressi come casi d'uso
- Ricavati:
  - Dalle specifiche date
  - Da interviste con il committente e gli utenti
- Mostrano le **funzionalità** che il sistema sistema dovrebbe fornire così come sono percepite da attori esterni
- Specifica le funzionalità del sistema (*use case*), chi le fruisce (*attori*) e le relazioni tra essi (*associazioni*)

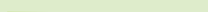

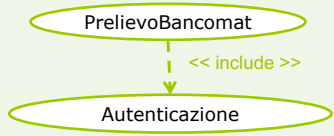
13

## Casi d'uso: Elementi

Costruttore	Descrizione	Sintassi
Caso d'uso	Funzionalità offerte dal sistema	
Attore	<ul style="list-style-type: none"> <li>• Utenti che interagiscono con il sistema</li> <li>• Sottinsieme software o componenti hardware con cui il sistema interagisce</li> </ul>	

14

## Casi d'uso: Relazioni (1)

Costruttore	Descrizione	Sintassi
<b>Associazione</b>	Relazione che unisce i casi d'uso agli attori	
<b>Inclusione</b>	<p>Indica che un caso d'uso è <b>incluso</b> nel caso d'uso base.</p> <p>Di solito si usa:</p> <ul style="list-style-type: none"> <li>quando un caso d'uso è complesso</li> <li>per descrivere un comportamento comune</li> </ul> <p>La freccia va dal caso d'uso che include (caso base) al caso d'uso incluso (caso incluso).</p>	<p></p> <p></p> <p>Durante il prelievo con il bancomat l'utente deve prima autenticarsi (inserendo oltre al bancomat anche il PIN).</p>

15

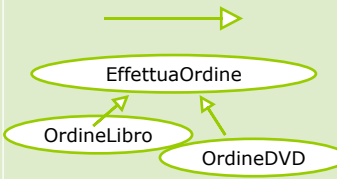
## Casi d'uso: Relazioni (2)

Costruttore	Descrizione	Sintassi
<b>Estensione</b>	<p>Indica che un caso d'uso estende il caso d'uso base.</p> <p>Si specificano quali punti vengono estesi rispetto al caso base.</p> <p>Modella sottoflussi <b>CONDIZIONALI</b>.</p> <p>La freccia va dal caso d'uso che estende al caso d'uso esteso.</p>	<p></p> <p></p> <p>Durante l'emissione dell'ordine, se si tratta di un ordine urgente, viene trattato come tale.</p>

16



## Casi d'uso: Relazioni (3)

Costruttore	Descrizione	Sintassi
<b>Generalizzazione</b>	Due casi d'uso offrono funzionalità simili, ma il caso d'uso figlio specializza quanto offerto dal padre (eredita e può estendere aggiungendo o modificando alcuni passi)	 <p>Durante l'ordine, l'utente può ordinare un libro o un DVD.</p>

Estensione e generalizzazione sono simili. L'estensione chiarisce quali punti siano effettivamente estesi.

17

## Casi d'uso vs Scenari

- Scenario (detto anche Storia o Narrativa): descrizione di un caso d'uso
  - Descrive **l'istanza di esecuzione** del caso base (o delle varianti principali) di un caso d'uso
  - Specifica anche le possibili eccezioni

Caso d'uso	.....
Attori	.....
Precondizioni	.....
Descrizione/Sequenza degli eventi	.....
Postcondizioni	.....
<b>Scenari alternativi</b>	.....
Note	.....

18

## Use Case Diagram: come procedere

1. Nel caso si abbia un testo che descrive i requisiti, come prima cosa identificare attori e casi d'uso nel testo, evidenziandoli (**ANALISI DEL TESTO**)
  1. Identificazione degli **attori** (utilizzatori del Sistema)
  2. Identificazione dei **casi d'uso** (tipi di uso del sistema, task, obiettivo specifico per l'attore o per il sistema)
2. Definizione delle **associazioni** fra attori e casi d'uso
- 3. Descrizione** dei casi d'uso (caso base e variante)

19

## Suggerimenti (1)

- UML Use Case Diagram: descrivono **l'interazione tra attori e Sistema**, **NON** la logica interna (*black box view*)
  - RICORDARSI A COSA SERVE QUESTA TIPOLOGIA DI DIAGRAMMA!
- **NB**: gli attori specificano un ruolo, corrispondono a una classe, non all'oggetto
- I dettagli interni vengono dati da:
  - Descrizione scenari
  - Diagrammi di interazione

20

## Suggerimenti (2)

- Non esagerare con l'uso di *inclusione*, *estensione* e *generalizzazione* (soprattutto nella fase di interazione con il committente)
- Spesso sono utili raffinamenti successivi



### Generalizzazione, estensione, inclusione

- Uso di **extend**
  - Usato per gestire le *eccezioni* o i casi **opzionali**
- **include**
  - Usato per gestire i *sottocasi* e i casi *comuni* con altri casi d'uso
- **generalizzazione**
  - Usato per gestire *casi generali* di cui si specificano casi particolari

21

## Use Case Diagram e Requisiti (1)

- Ogni caso d'uso può soddisfare **più requisiti**
- Un requisito può dare origine a **più casi d'uso**
- Ad ogni caso d'uso possono venire associati **più requisiti**

22

## Use Case Diagram e Requisiti (2)

### I casi d'uso servono a:

- chiarire i requisiti del committente in termini comprensibili
- trovare aspetti comuni (riuso)
- individuare gli attori del sistema
- individuare gli eventi a cui il sistema deve rispondere

23

## Caso di studio: Garage sotterraneo

- Esercizio finale tratto dal libro: UML Distilled, M. Fowler 4th edition
- Su Ariel trovate il testo che descrive informalmente in linguaggio naturale i requisiti
- **IN BREVE:** Si modelli con diagrammi di casi d'uso il software di gestione di un garage sotterraneo con 200 posti auto.

24

**Fase 1****Analisi del testo:**

1. Identificazione degli **attori** (utilizzatori del Sistema)
2. Identificazione dei **casi d'uso** (tipi di uso del sistema, task, obiettivo specifico per l'attore o per il sistema)

25

**Fase 2****Iniziare ad abbozzare il diagramma**

- Intro a StarUML

26