



Modellazione con UML



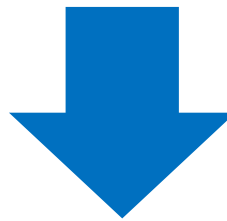
Modelli

Cos'è un modello?

- Un modello è una rappresentazione semplificata della realtà

Perché creare modelli?

- Perché la realtà può essere troppo complessa



- I modelli aiutano a **comprendere** e a **comunicare**



Punti di vista sulla modellazione

Prospettiva concettuale

- Rappresentazione di concetti caratteristici del dominio studiato
 - domain model
- È indipendente dal software

Prospettiva software

- Gli elementi di un modello corrispondono agli elementi di un sistema software
 - prima che il sistema sia sviluppato: **forward engineering**
 - partendo da un sistema già esistente: **reverse engineering**

Cos'è UML

(Unified Modeling Language)



- E' una famiglia di notazioni grafiche per la **modellazione visuale del software**
 - È usato principalmente per l'analisi e la progettazione orientata agli oggetti (OOA – OOD)
- Come viene usato
 - **come abbozzo (sketch)**
 - come design tecnico dettagliato (blueprint)
 - come linguaggio di programmazione





UML come abbozzo

- Lo scopo principale è **favorire la comprensione e la comunicazione nelle discussioni**
- Usato per aiutare a **concentrarsi su alcuni aspetti** di un sistema software
- Criteri fondamentali
 - **Selettività**
 - Solo alcuni aspetti del sistema software sono modellati graficamente
 - Qualsiasi informazione può essere soppressa: l'assenza di qualcosa non significa che non esista
 - **Espressività**
 - Diagrammi intesi come figure
- I diagrammi sono creati improvvisando e in modo collaborativo (anche con uso di lavagne)

UML come design tecnico dettagliato



- Lo scopo principale è fornire ai **programmatore uno schema da seguire** (blueprint)
 - Approccio ispirato ad altre branche dell'ingegneria
- Usato per **guidare e documentare** la realizzazione di un sistema software
- Criteri fondamentali
 - **Completezza**
 - **Non ambiguità**
- I diagrammi creati sono figure che fanno parte della documentazione del sistema

UML come linguaggio di programmazione



- Lo scopo principale è permettere agli sviluppatori di **programmare in modo visuale**, indipendentemente dalla piattaforma software adottata
- Usato per **compilare direttamente i diagrammi in formato eseguibile**
 - Model Driven Development (MDD): nessuna distinzione tra forward e reverse engineering



Diagrammi UML

- **Diagramma strutturale** (structure diagram)
 - Diagramma delle classi (class diagram)
 - Diagramma dei componenti (component diagram)
 - Diagramma di struttura composita (composite structure diagram)
 - Diagramma di deployment (deployment diagram)
 - Diagramma degli oggetti (object diagram)
 - Diagramma dei package (package diagram)
- **Diagramma comportamentale** (behavior diagram)
 - Diagramma delle attività (activity diagram)
 - Diagramma dei casi d'uso (use case diagram)
 - Diagramma di macchina a stati (state machine diagram)
 - **Diagramma di interazione** (interaction diagram)
 - Diagramma di sequenza (sequence diagram)
 - Diagramma di comunicazione (communication diagram)
 - Diagramma di interazione generale (interaction overview diagram)
 - Diagramma di temporizzazione (timing diagram)



Diagramma delle classi

- Mostra le classi, le loro caratteristiche e le relazioni
- E' il diagramma più noto
- Molto utilizzato sia per modelli con prospettiva concettuale che per modelli con prospettiva software

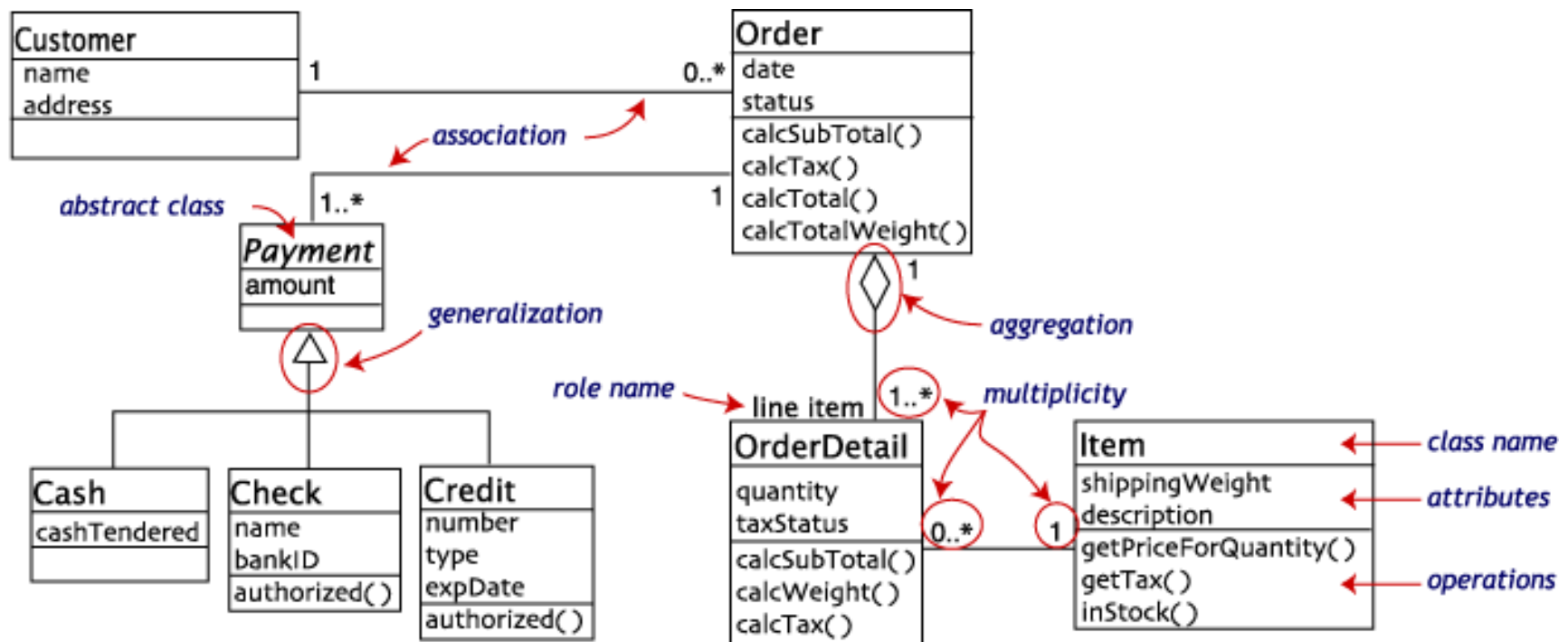




Diagramma dei componenti

- Mostra i componenti di un sistema e le loro connessioni

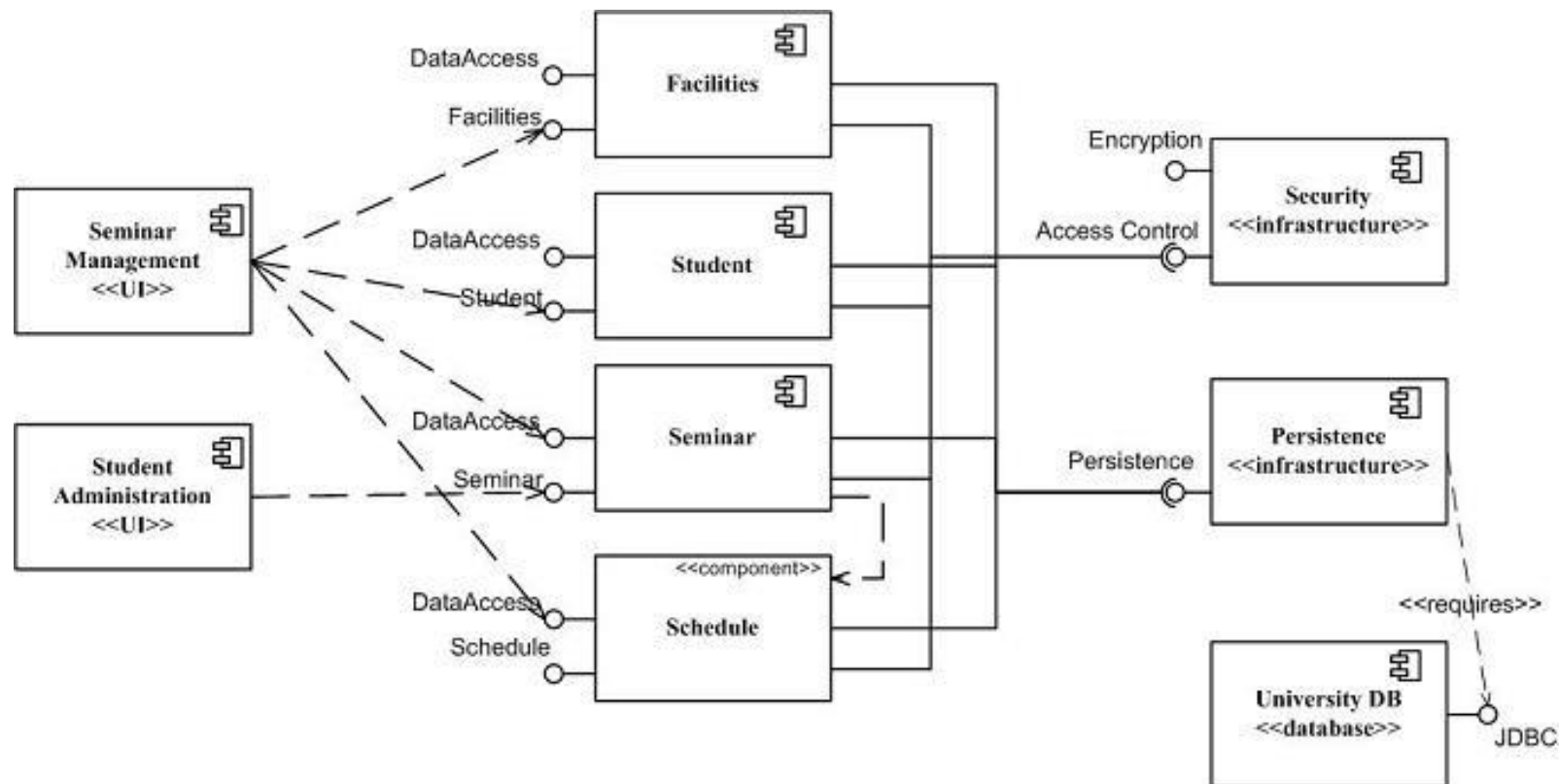


Diagramma di struttura composita



- Mostra la scomposizione di un classificatore (caso d'uso, classe, componente) a runtime
- Introdotto con UML 2
- Utilizzato per mostrare in modo esplicito le realizzazioni di casi d'uso

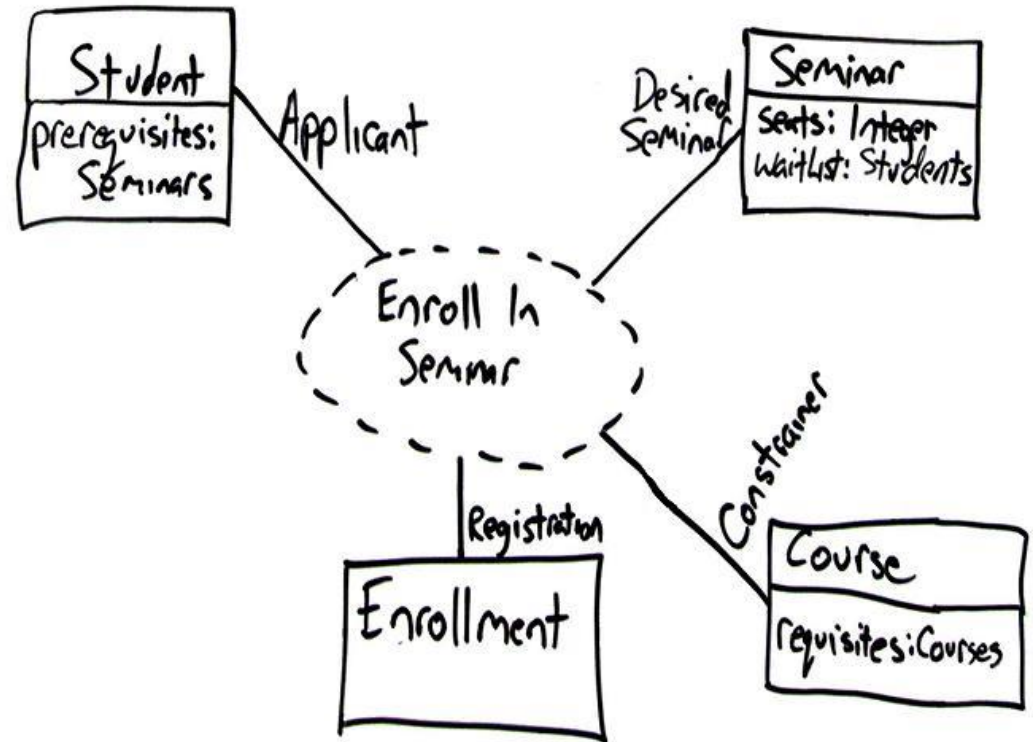




Diagramma di Deployment

- Mostra la distribuzione dei componenti software nei diversi nodi di elaborazione
- Utilizzato quando la “messa in opera” è complessa

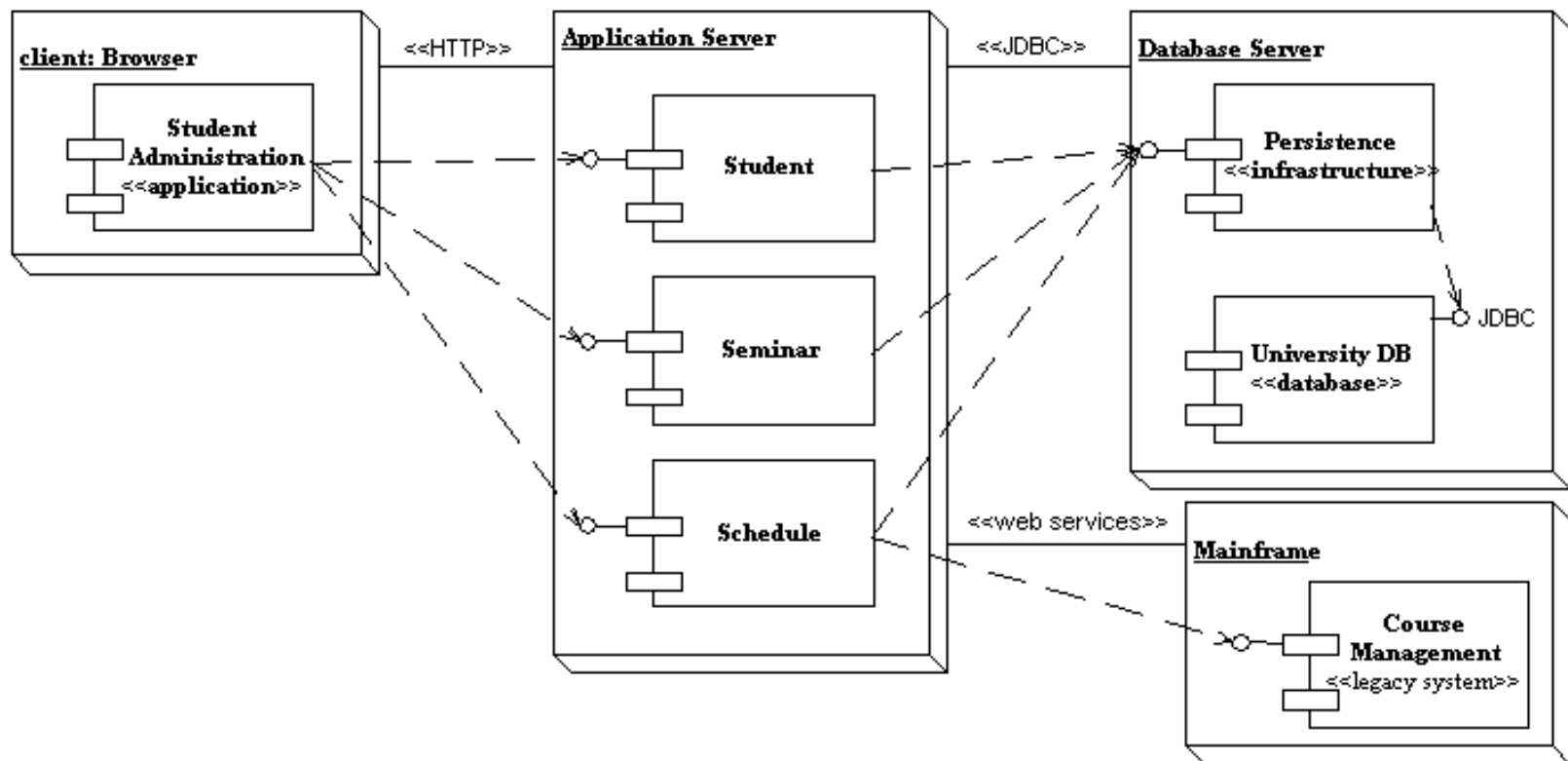




Diagramma degli oggetti

- Mostra una configurazione di istanze di classi e dei loro collegamenti in un certo istante di tempo

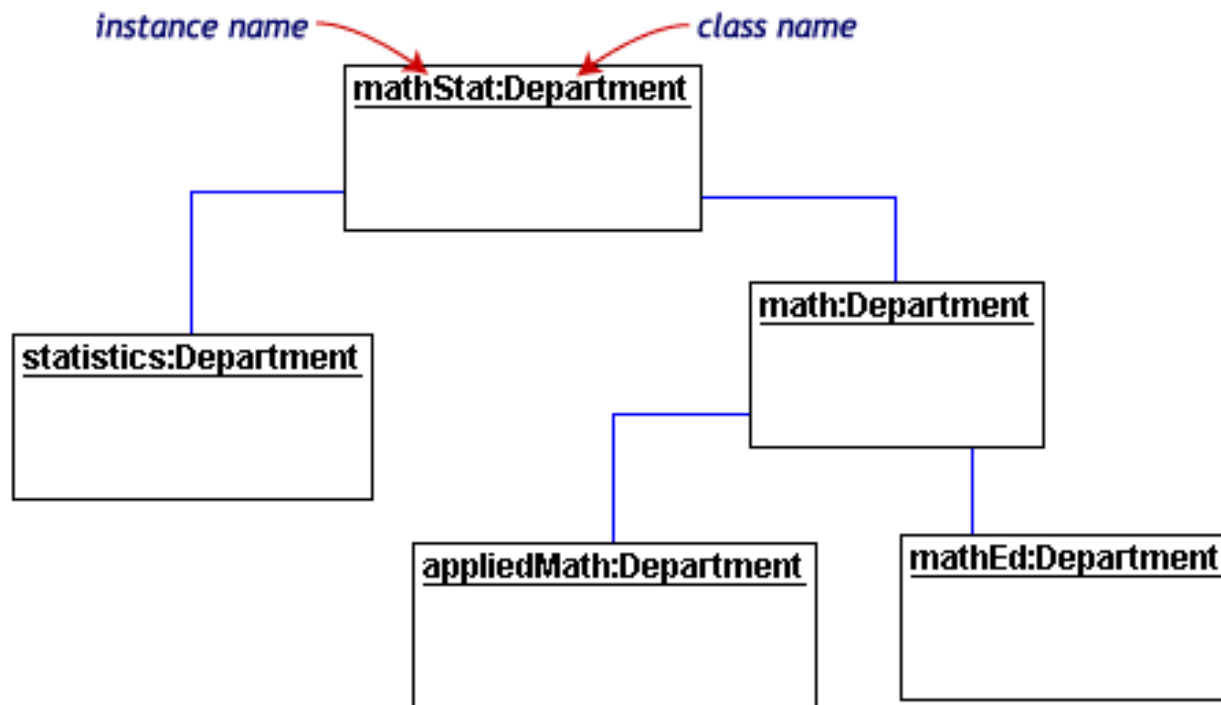


Diagramma dei package

- Mostra i package e le dipendenze tra di essi
 - Un package è un costrutto di raggruppamento
 - di classi, casi d'uso, componenti
- Utilizzato per visualizzare le dipendenze nei progetti di sistemi di grande scala

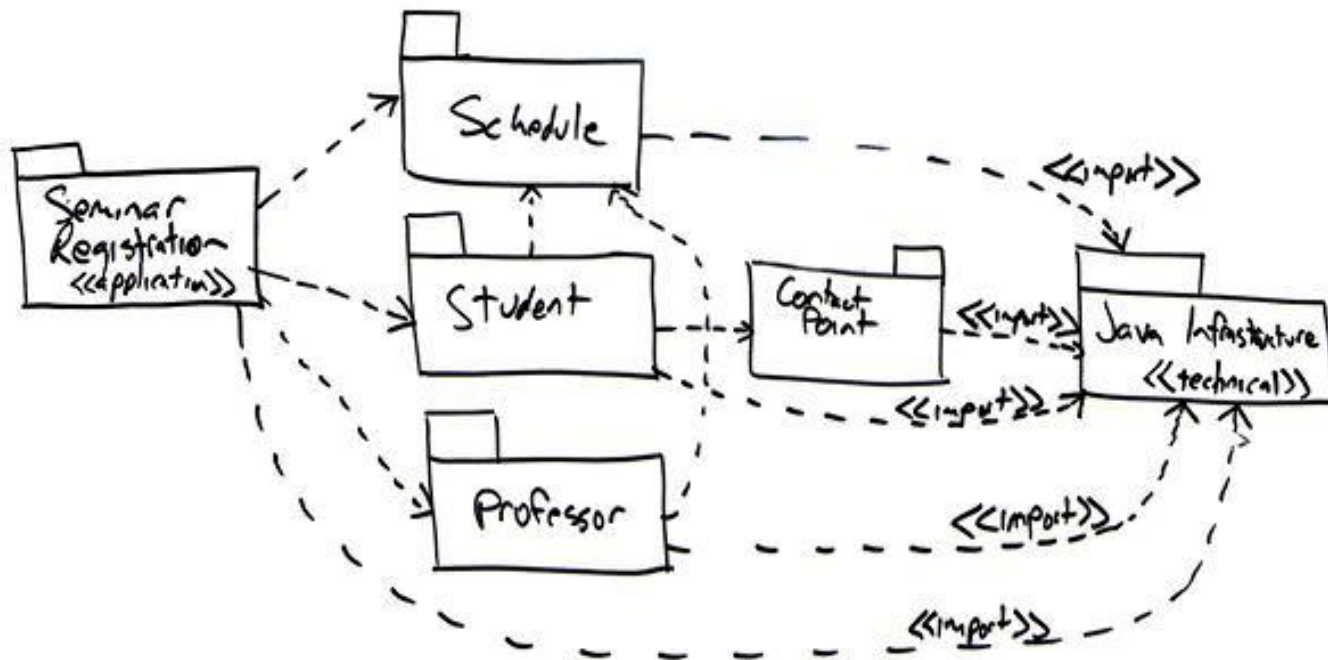




Diagramma delle attività

- Mostra il comportamento procedurale e parallelo
- Utilizzato per la modellazione di processi e workflow

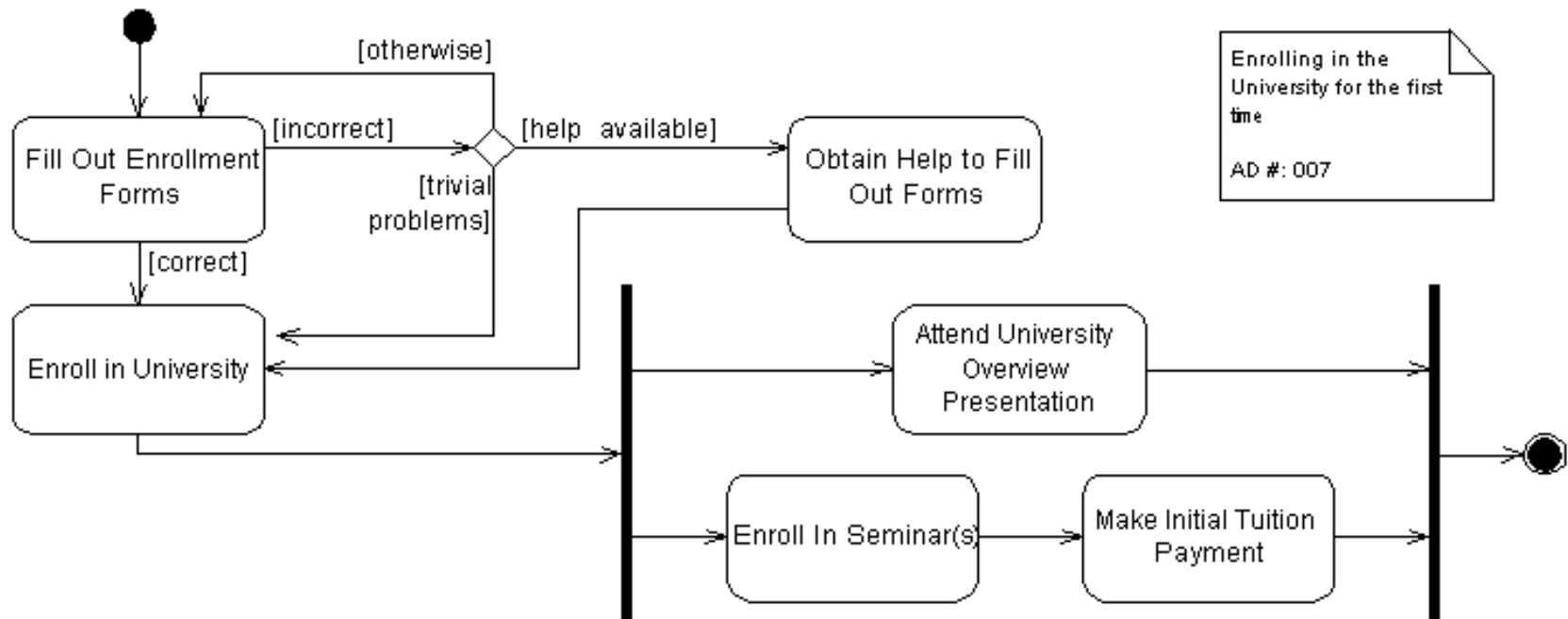




Diagramma dei casi d'uso

- Mostra come gli utenti interagiscono con un sistema
- Già presente in UML 1
- Il diagramma non è così importante come lo sono le descrizioni dei casi d'uso

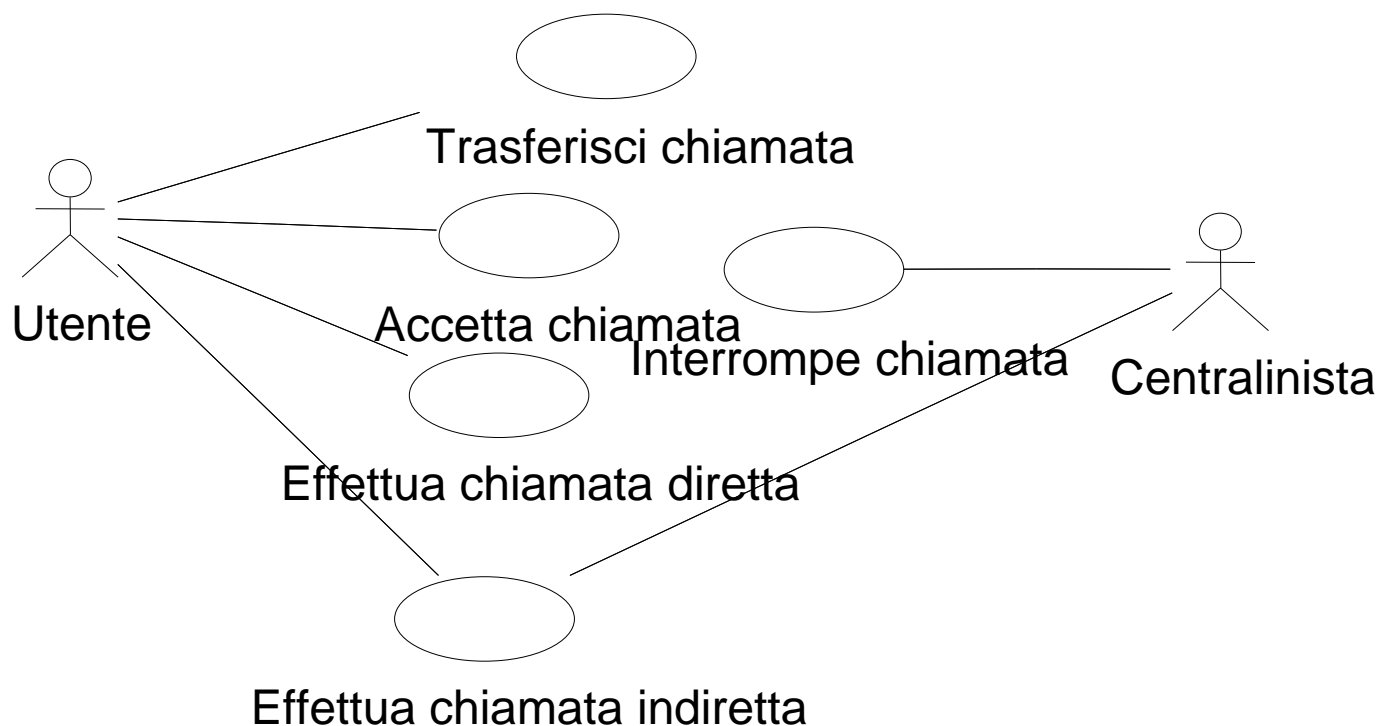




Diagramma di macchina a stati

- Mostra come gli eventi cambiano un oggetto durante il suo ciclo di vita
- Utilizzato per le classi che hanno una logica complessa
 - In genere oggetti di controllo (controller o monitor) e oggetti reattivi (interfacce grafiche, sensori)

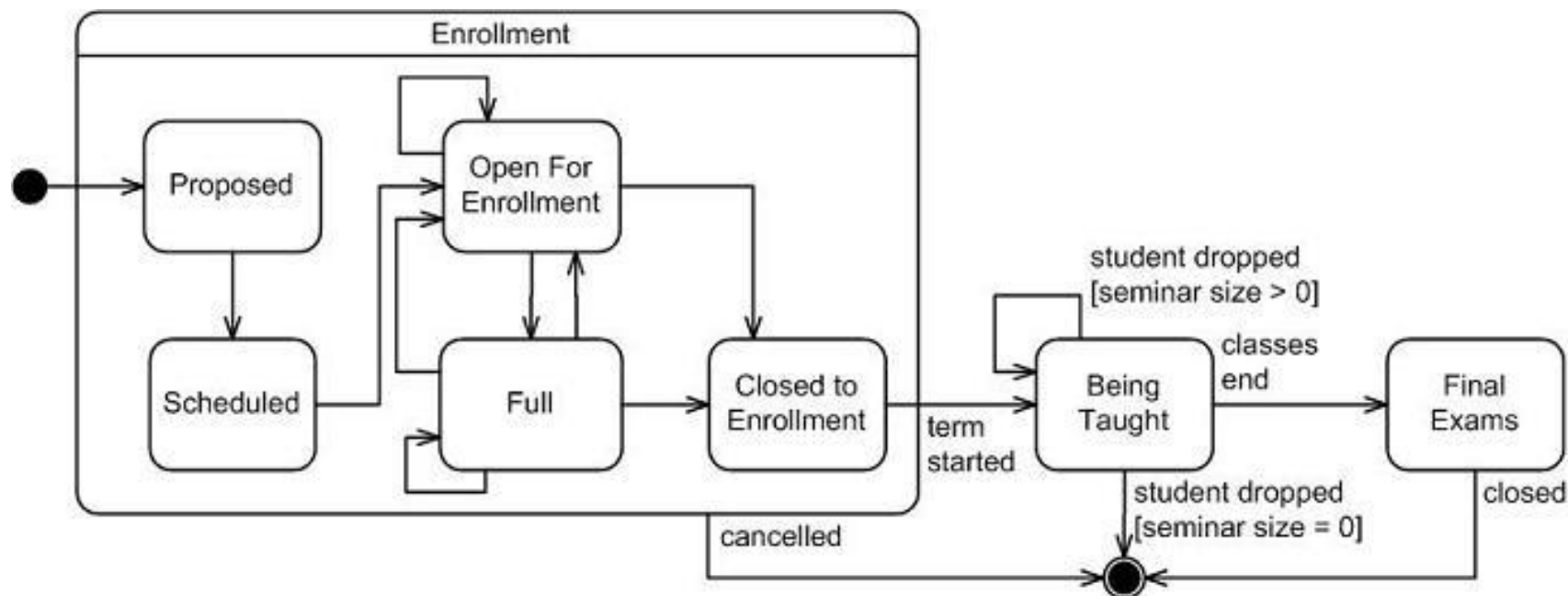




Diagramma di sequenza

- Mostra l'interazione tra oggetti con enfasi sulla sequenza dei messaggi
- Molto utilizzato per rappresentare il comportamento di un insieme di oggetti in uno scenario di caso d'uso (funzionalità)

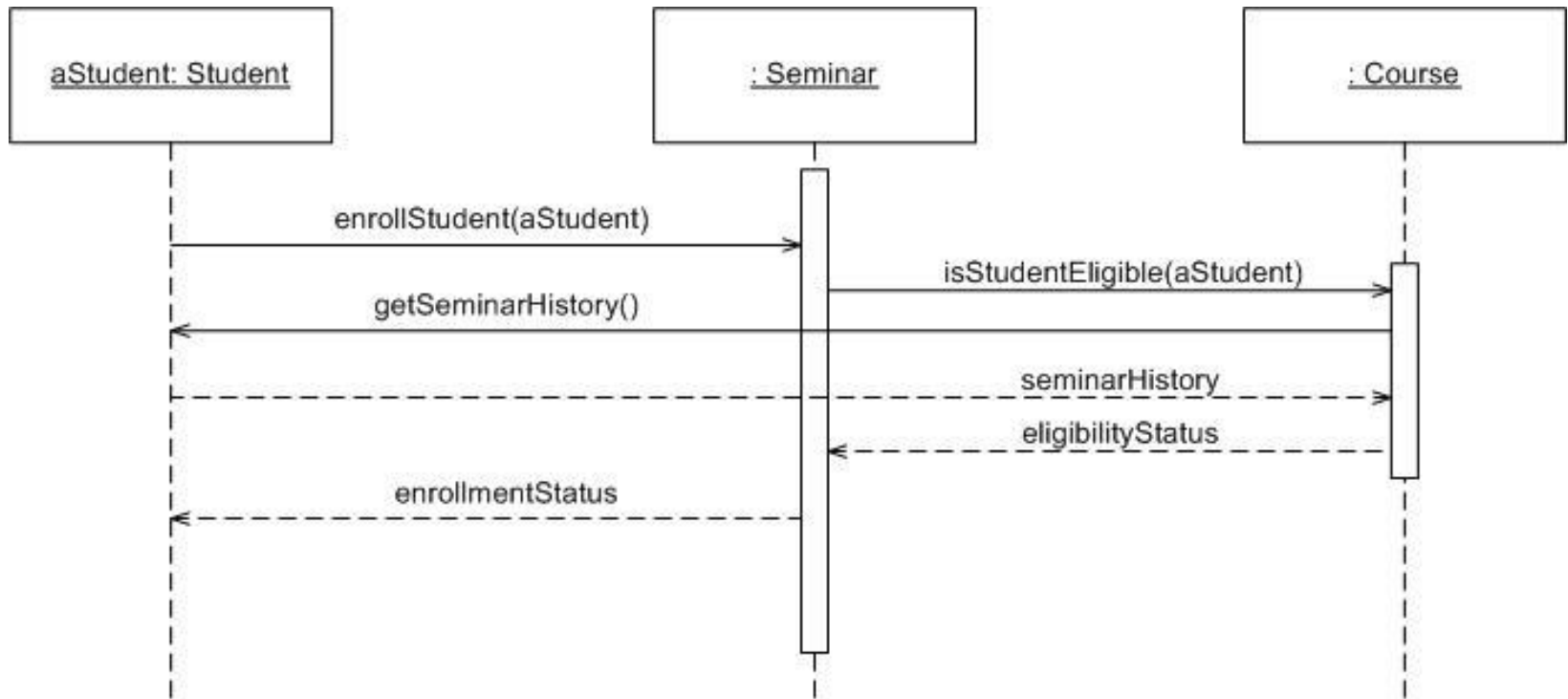




Diagramma di comunicazione

- Mostra l'interazione tra oggetti con enfasi sui collegamenti
- E' equivalente ai diagrammi di sequenza, ma meno utilizzato

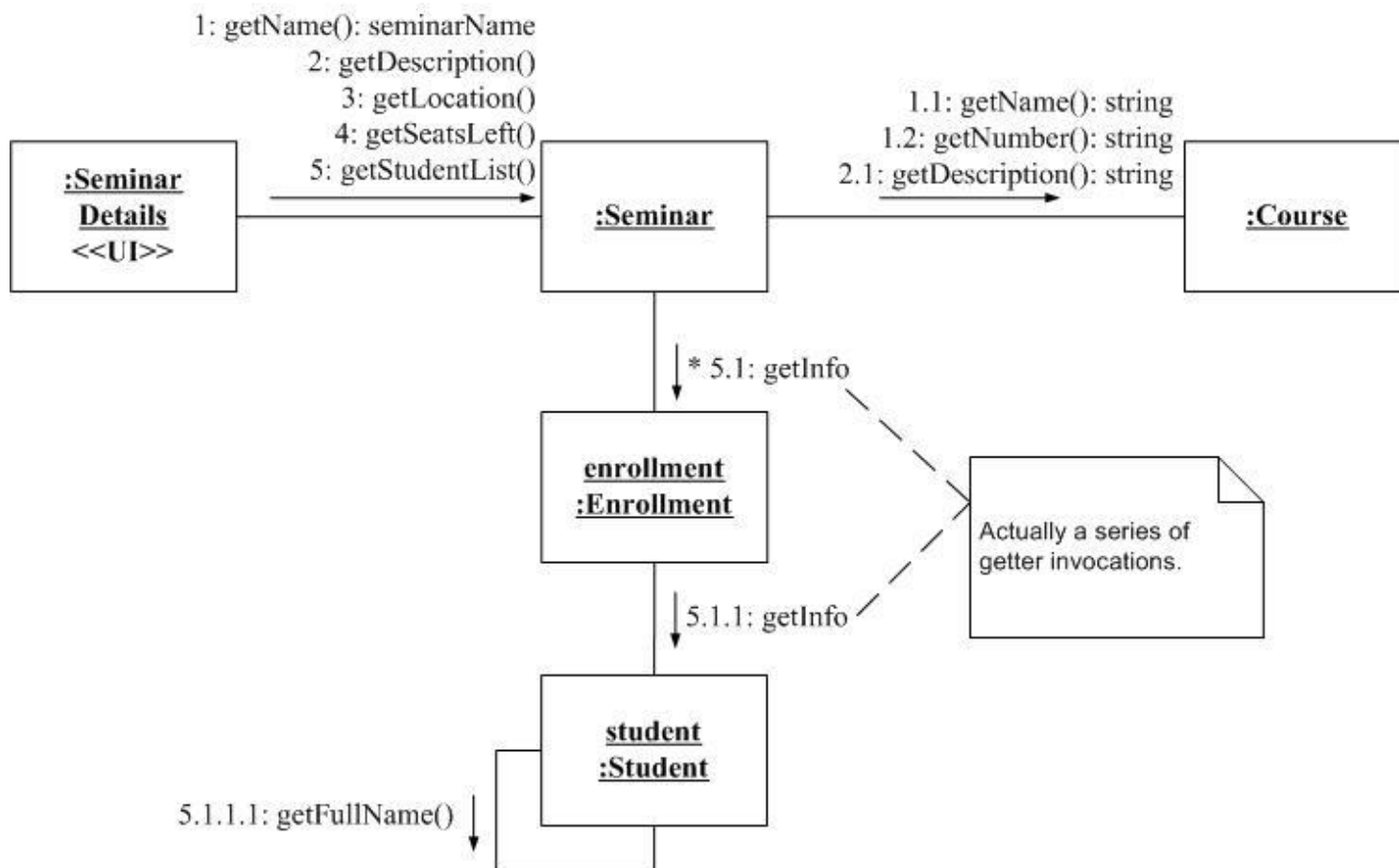




Diagramma di interazione generale

- Fusione di un diagramma di sequenza con un diagramma delle attività
- Introdotto con UML 2

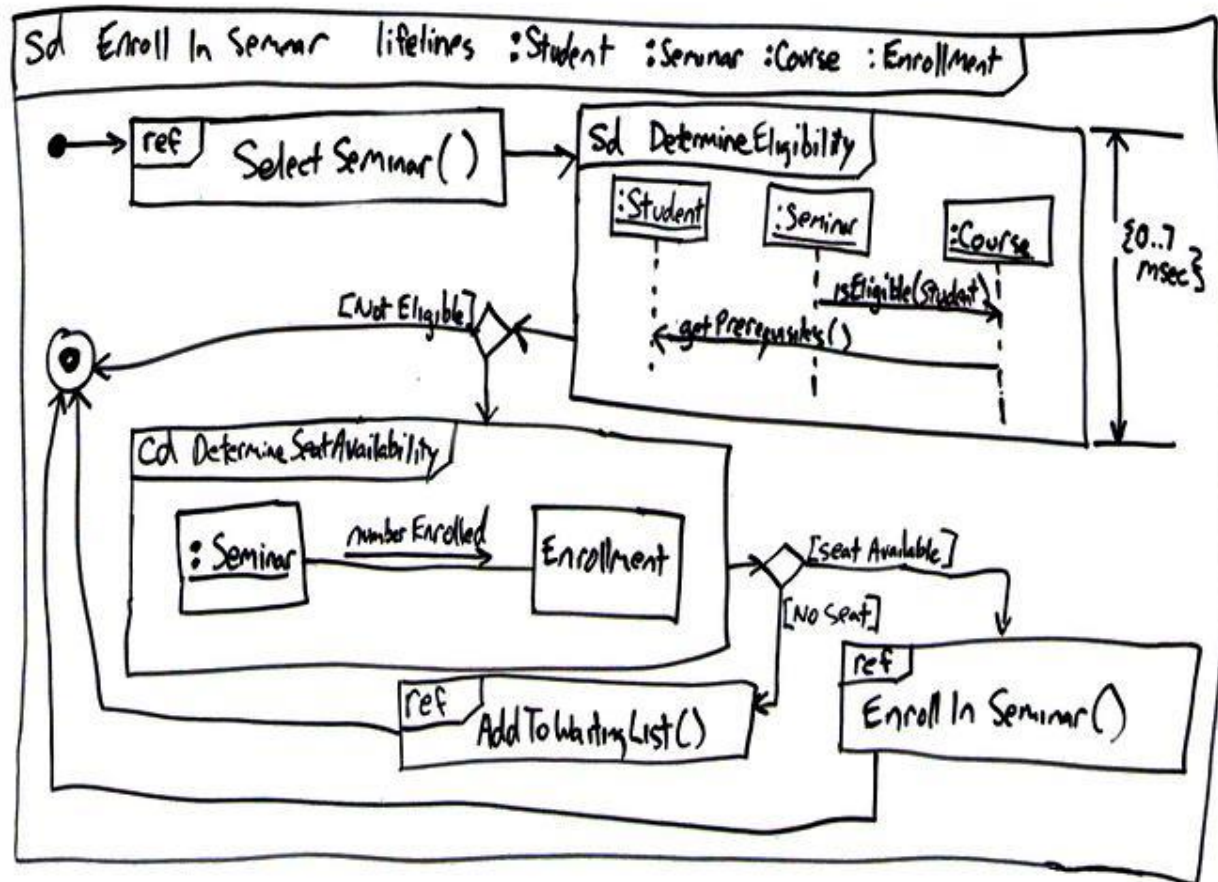
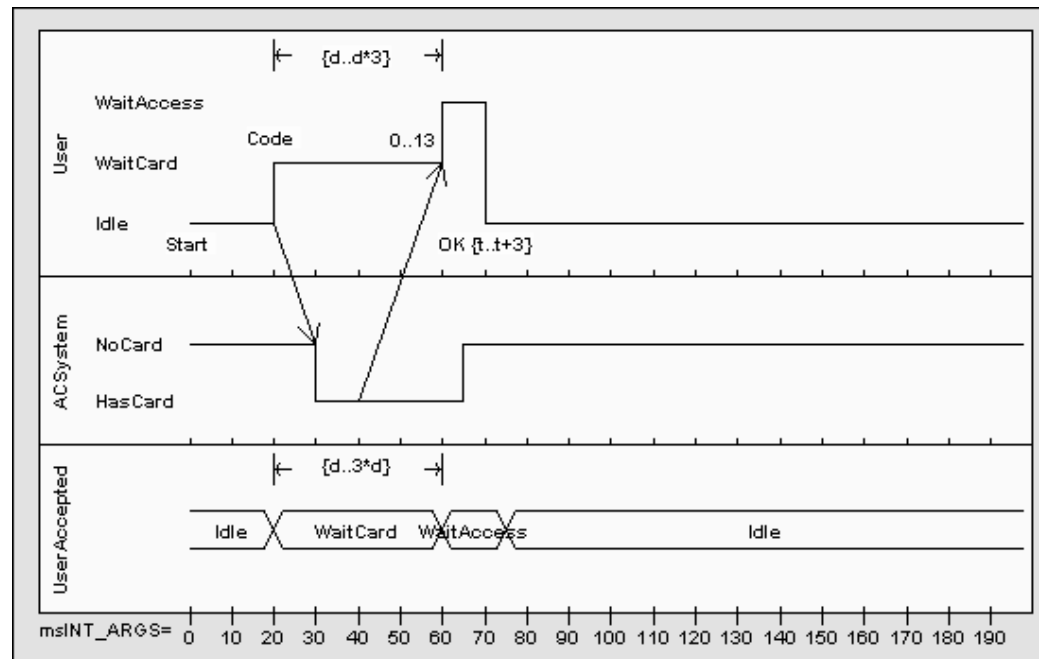




Diagramma di temporizzazione

- Mostra l'interazione tra oggetti con enfasi sul tempo
- Introdotto con UML 2
- Utilizzabile per software embedded e real-time
- Molto utilizzato dagli ingegneri elettronici per la progettazione di componenti elettronici



Diagrammi UML solitamente utilizzati per l'attività di analisi dei requisiti



- Diagramma dei casi d'uso
 - Piuttosto inutile
- **Diagramma delle classi**
 - con prospettiva concettuale (modello del dominio)
- ***Diagramma delle attività***
 - Solo se è utile mostrare il flusso di lavoro nell'organizzazione, evidenziando “chi fa che cosa”
- Diagramma di macchina a stati
 - Solo se il sistema da realizzare ha un comportamento complesso, che cambia al verificarsi di determinati eventi

Diagrammi UML solitamente utilizzati per l'attività di design



- **Diagrammi delle classi**
 - con prospettiva software
 - solo per i package o per i casi d'uso più importanti (UML come abbozzo)
- **Diagrammi di sequenza**
 - con prospettiva software
 - solo per gli scenari più importanti dei casi d'uso (UML come abbozzo)
- **Diagrammi di macchina a stati**
 - solo per le classi che hanno un ciclo di vita complesso (UML come abbozzo)
- **Diagramma dei package**
 - solo se il software è su larga scala e ha un'architettura complessa
- **Diagramma di deployment**
 - solo se il software è distribuito e si ritiene utile visualizzare i protocolli di comunicazione e come i componenti sono distribuiti tra i diversi nodi

How the UML diagram describes the software



How the code is actually written

