

Sim
UniPisa
LaSpezia

UML per la modellazione discreta

Simulazione – Lezione n. 6
Corso di Laurea in Informatica Applicata
Università di Pisa, sede di La Spezia

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 1/24 

Sim
UniPisa
LaSpezia

Contenuti

- UML in poche parole
- UML per la modellazione discreta
- Un metodo di modellazione basato su UML
- Eventi, entità & OO (e un pizzico di agenti)
- Esempio: l'impiegato tormentato con UML

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 2/24 

Sim
UniPisa
LaSpezia

UML in breve

- Cosa è
 - È un linguaggio di modellazione
 - È nato ed è pensato per la modellazione di sistemi software
 - È versatile e si adatta all'uso anche in altri contesti
 - È standard, è diffuso ed è importante conoscerlo
- Cosa non è
 - Non è un metodo di modellazione
 - Non rende la modellazione facile
 - Non da nessuna primaria qualità alla modellazione
 - Non è (ancora) lo standard per la modellazione
 - Non è standard il modo di usare UML

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 3/24 

Sim
UniPisa
LaSpezia

Storia di UML

- Metodi per lo sviluppo orientato agli oggetti
 - OOAD (Booch), OOSE (Jacobson), OMT (Rumbaugh)
 - OOPSLA '95: proposta di un *Unified Method* (0.8)
- Dal metodo alla notazione
 - UML 0.9, Rational & Partners (giu. 1996)
 - UML 1.0-1.1, OMG (gen.-set. 1997)
 - UML 1.5, ISO/IEC19501 (2001/2002)
 - UML 2.0, (ago. 2004)
 - ...
 - UML 2.4.1, (ago. 2011)

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 4/24

Sim
UniPisa
LaSpezia

Stato di UML

- Object Management Group
 - Consorzio per la promozione di standard
 - CORBA, ..., UML, XMI, MDA, ... SysML, ...
- Organizzazione della specifica di UML (da 2.0)
 - Infrastruttura: elementi del linguaggio
 - Sovrastruttura: linguaggio (definito sull'infrastruttura)
- Uso ed evoluzione
 - Estensioni: tendono a diventare standard
 - Dialetti: sono prassi riconosciute, ma non ufficiali
 - Uso conforme, convenzionale, non-conforme
 - Evoluzioni naturali e previste (stereotipi, profili, ...)

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 5/24

Sim
UniPisa
LaSpezia

Un linguaggio per diagrammi

```
graph TD; Diagram[Diagram] --> Structure[Structure Diagram]; Diagram --> Behavior[Behavior Diagram]; Structure --> Class[Class Diagram]; Structure --> Component[Component Diagram]; Structure --> Object[Object Diagram]; Behavior --> Activity[Activity Diagram]; Behavior --> UseCase[Use Case Diagram]; Component --> Profile[Profile Diagram]; Component --> Composite[Composite Structure Diagram]; Component --> Deployment[Deployment Diagram]; Component --> Package[Package Diagram]; Object --> Interaction[Interaction Diagram]; Object --> StateMachine[State Machine Diagram]; Interaction --> Sequence[Sequence Diagram]; Interaction --> Communication[Communication Diagram]; Interaction --> Overview[Interaction Overview Diagram]; Interaction --> Timing[Timing Diagram];
```

Notation: UML

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 6/24

Sim
UniPisa
LaSpezia

Come si dovrebbe usare UML?

- Problemi nell'uso di UML
 - La dimensione del linguaggio produce incertezza
 - Comprenderlo completamente è costoso
 - Soffre l'abitudine all'uso di notazioni precedenti
 - Informali (flowcharts, CRC cards ...)
 - Formali (diagrammi E-R, reti di Petri ...)
- Modi di usare UML (riconosciuti)
 - Bozze (sketch), per discutere
 - Disegni (blueprint), per progettare
 - Programmi (code), per generare codice

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 7/24 

Sim
UniPisa
LaSpezia

Modellazione per la simulazione

- Obiettivo
 - Specificare il modello di un sistema
 - Procedendo per gradi, per parti e relazioni
 - Utilizzando una notazione
 - Arrivando bene e presto all'implementazione del simulatore
- Requisiti di un linguaggio di modellazione
 - Fornire una sintassi per diversi "aspetti" della modellazione
 - Facilitando analisi e ragionamento ...
 - ... comunicazione e documentazione ...
 - ... fino a permettere traduzione o interpretazione

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 8/24 

Sim
UniPisa
LaSpezia

Domini di modellazione

- UML e ingegneria del software
 - Modellazione di sistemi software
 - Analisi, specifica, progettazione
- Modellazione con UML per la simulazione
 - Modellazione di sistemi dinamici discreti, generici
 - Simulati con sistemi software
- Corrispondenze
 - Sistema fisico
 - Modello UML del sistema fisico
 - Software del simulatore derivabile dal modello

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 9/24 

Sim
UniPisa
LaSpezia

Dai cicli di attività a UML

- I cicli di attività hanno una tradizione
 - Conservare alcuni aspetti del metodo
 - Specifica basata sulla definizione di stati e transizioni
 - Mantenere un certo grado di corrispondenza
 - Aumentare la corrispondenza con il sistema fisico
- Diagrammi delle macchine a stati
 - Eventi, guardie, azioni (fatti condizionati con conseguenze)
 - Per ogni entità (specificati per classe, eseguiti dalle istanze)
- Diagrammi delle classi, degli oggetti, delle attività
 - Per dare sostanza alle informazioni nei DMS
 - Attributi, metodi, istanze, sequenze stabilite

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 10/24 

Sim
UniPisa
LaSpezia

Aspetti della modellazione

- Stato, entità e insiemi
 - Classi, tipi di dato astratto
 - Attributi, i valori caratterizzano le istanze delle classi
 - Stati, caratterizzano il comportamento di un'entità
 - Oggetti, istanze delle classi
 - Insiemi, collezioni di oggetti (con varie politiche)
- Operazioni, eventi, attività, processi
 - Eventi, fatti che avvengono a dati istanti
 - Condizioni, da verificare sullo stato (di un oggetto)
 - Azioni, conseguenze di un evento (su un oggetto)
 - Politiche, regole di propagazione degli eventi agli oggetti

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 11/24 

Sim
UniPisa
LaSpezia

Metodo di modellazione - 1

- Identificare le entità (UML: classi)
 - Attive, reagiscono agli eventi
 - Passive (dati), utili solo a rappresentare dati
- Identificare gli insiemi (UML: classi, oggetti)
 - Risorse, politiche
- Identificare gli eventi (UML: classi)
 - Interni, sono generati e programmati da entità del sistema
 - Esterni, sono generati esternamente al sistema
- Identificare le istanze delle entità (UML oggetti)
 - Dare sostanza al sistema

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 12/24 

- Attributi e metodi
- Degli eventi (UML: classi e attività)
 - Attributi ereditati: tempo, priorità
 - Specializzazione: doAction(), propagazione agli oggetti
 - Eventuali dati e metodi dipendenti dal sistema da modellare
- Delle entità (UML: classi e macchine a stati)
 - Dati e metodi dipendenti dal sistema da modellare
 - Specializzazione handle(e: Event)
 - Come macchina a stati
- Procedere per raffinamenti successivi

- Metodo ad eventi e agenti, orientato agli oggetti
 - Classi di basi da specializzare, eventi ed entità
 - Con un pizzico di agenti: oggetti multipli e autonomi
- Nella modellazione
 - Come strumento di analisi, per rendere concreti i concetti
 - Concretezza sintattica ai pezzi da cercare
- Nella realizzazione del simulatore
 - Accesso al motore (programmazione, tempo corrente)
 - Code e entità incodabili (in generale insiemi)
 - Schemi di derivazione del codice
 - Metodo legato a uno strumento (libreria C++)

- Formulazione originale
 - Alcuni clienti arrivano allo sportello altri telefonano
 - Lo stesso impiegato serve entrambi i tipi di clienti
 - I clienti allo sportello aspettano in coda
 - Il centralino mette in attesa le chiamate
 - I clienti allo sportello hanno sempre la priorità
 - I tempi di arrivo/chiamata e servizio sono dati dei clienti
 - Le code e la pazienza dei clienti sono infinite (assunzione)
- Seguire il metodo
 - Valore delle precedenti modellazioni: esperienza
 - Confronto fra metodi, non riuscito dei risultati

Sim
UniPisa
LaSpezia

Modello concettuale

- Entità
 - Un impiegato
 - Molti clienti, due classi: *sportello e telefono*
 - Possibili specializzazioni di un generico cliente (OO ci piace)
 - Tutte entità attive (è una scelta, non obbligata)
- Insiemei
 - Due code, una per tipo (sono fisiche: c'è una politica)
- Eventi
 - Arrivo di un cliente allo sportello esterno
 - Chiamata di un cliente al telefono esterno
 - Fine di un servizio, di sportello o al telefono interno

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 16/24

Sim
UniPisa
LaSpezia

Classi e oggetti per entità e insiemei

- Componenti del sistema, diagramma degli oggetti
 - Nomi delle classi
 - Istanze delle classi
- Classi
 - Clerk
 - DClient
 - PClient
- Insiemei
 - Code allo sportello e al telefono
 - Sperabilmente istanze di un tipo predefinito

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 17/24

Sim
UniPisa
LaSpezia

Classi e attività per gli eventi

- Tre eventi, tre classi
 - Arrivo allo sportello
 - Chiamata al telefono
 - Fine di un servizio
- Specializzazione
 - Priorità
 - Riferimenti alle entità alle quali gli eventi devono essere propagati
 - Metodo doAction() sequenza di propagazione

Giovanni A. Cignoni – Simulazione – www.di.unipi.it/~giovanni 18/24

Sim
UniPisa
LaSpezia

Macchine a stati: i clienti

- Specifica di handle(e: Event) per i clienti
- Un solo stato
 - In servizio, può darsi in attesa
 - Stato iniziale e transizione esplicita
 - Stato finale, il cliente termina
- Soluzioni particolari
 - I clienti si mettono sempre in coda
l'ordine della propagazione nella doAction() è importante
 - Priorità degli eventi
 - garantisce la politica in arrivi/chiamate contemporanee
 - evita inutili transizioni (coda vuota, arrivo insieme a fine)

Giovanni A. Cignoni - Simulazione - www.di.unipi.it/~giovanni 22/24

Sim
UniPisa
LaSpezia

Considerazioni

- Entità attive e passive
 - Passive: dati, che qualcuno deve gestire
 - Scelta per attive: tendenza a maggior disaccoppiamento
- Hardboiled OO
 - Fattorizzazione del metodo handle
 - Soluzione elegante, non sempre necessaria
 - A volte poco riusabile (clienti con comportamento diverso)
- Generazione eventi esterni
- Derivazione del codice
- Variazioni con riuso di modelli e codice

Giovanni A. Cignoni - Simulazione - www.di.unipi.it/~giovanni 23/24

Sim
UniPisa
LaSpezia

Riferimenti

- G. Gallo, *Note di Simulazione*, cap. 2
- UML: Infrastructure & Superstructure 2.3
Object Management Group
- M. Fowler, "UML Distilled", Addison-Wesley

Giovanni A. Cignoni - Simulazione - www.di.unipi.it/~giovanni 24/24

