



UML

Il linguaggio UML e ArgoUML

30/09/2009

Ingegneria dei sistemi software
2009/2010
manuel.comparetti@iet.unipi.it



UML

- Unified Modeling Language
 - una famiglia di notazioni grafiche standardizzate* orientata agli oggetti
 - si usa per descrivere e progettare sistemi software, in particolare (ma non necessariamente) utilizzando paradigmi object oriented

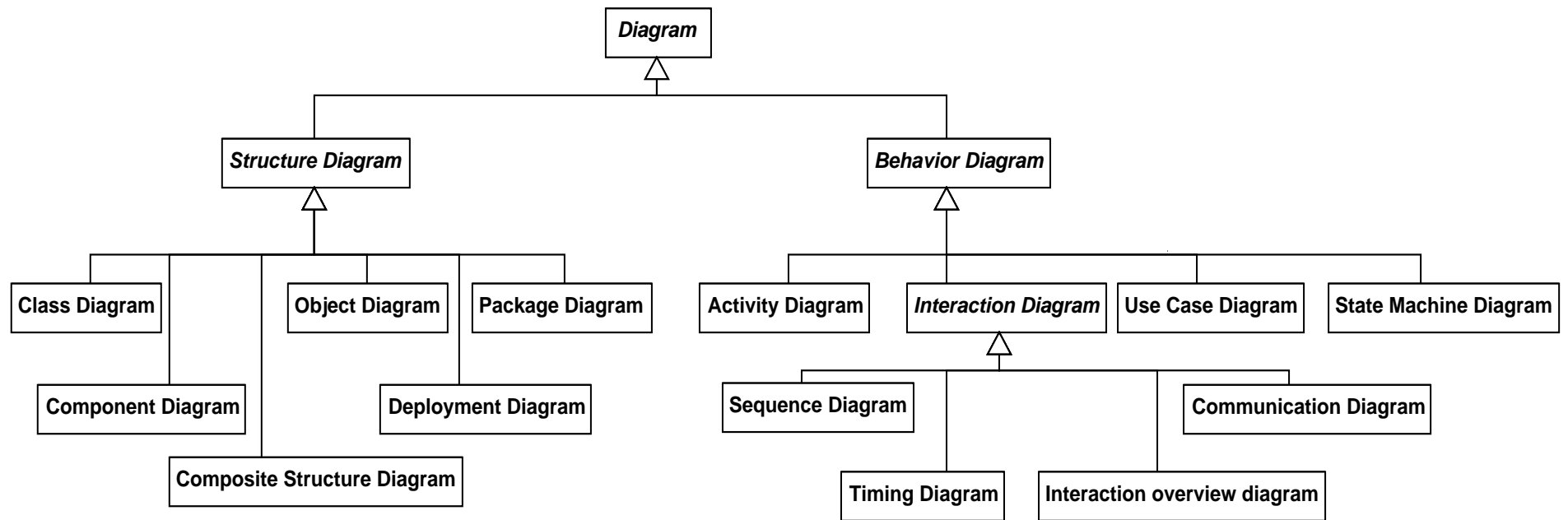
*<http://www.omg.org/>



UML e viste

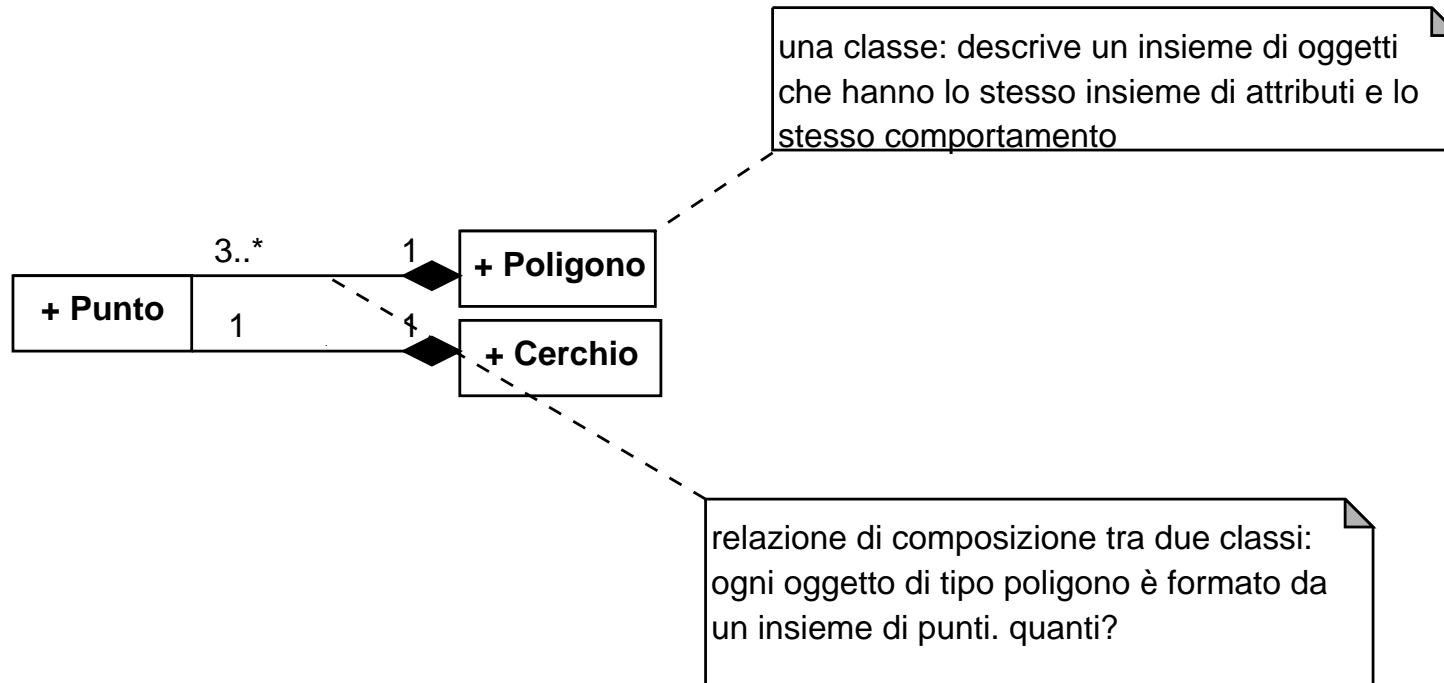
- In UML, un sistema viene descritto utilizzando diverse **viste** (views), e le rappresentazioni corrispondenti sono realizzate da diversi tipi di **diagrammi**
- Ogni **diagramma** è formato da rappresentazioni grafiche che descrivono un particolare aspetto del sistema
- chiameremo **Modello** un insieme di diagrammi che descrivono aspetti diversi dello stesso sistema

Tipi di Diagrammi UML



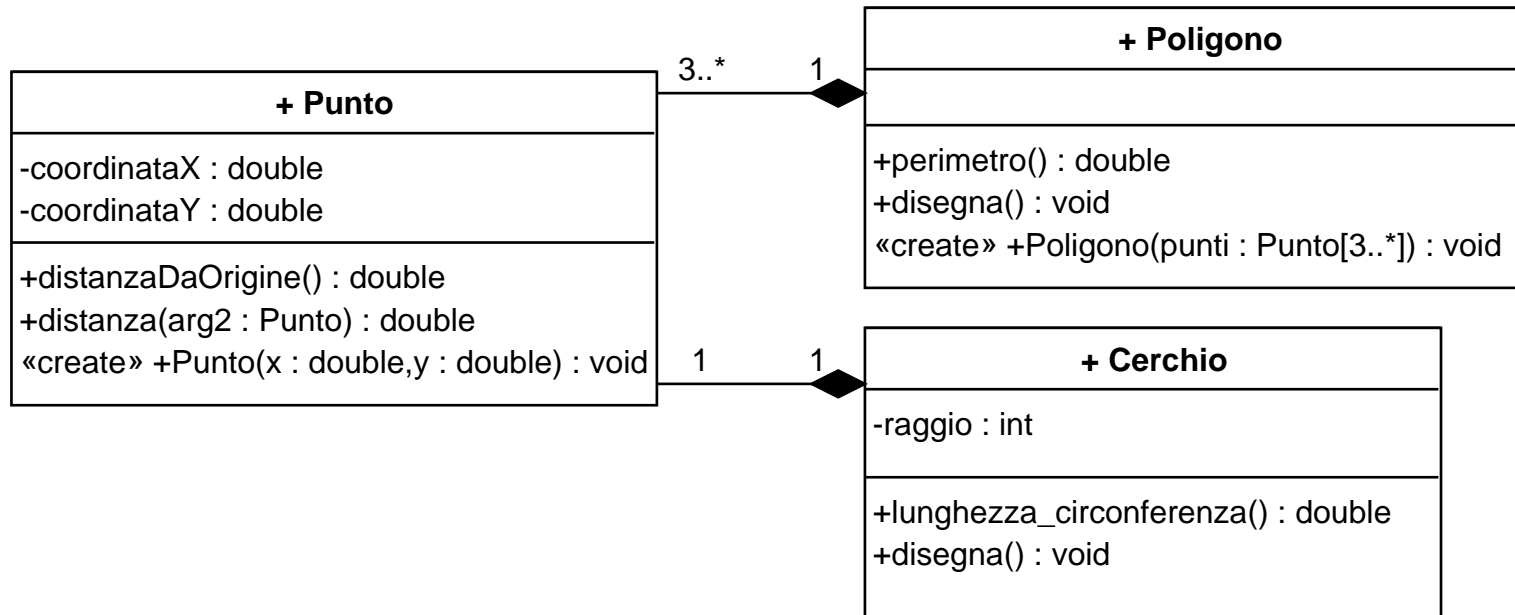
- UML 2 comprende 13 tipi di diagrammi

Esempio: class diagrams



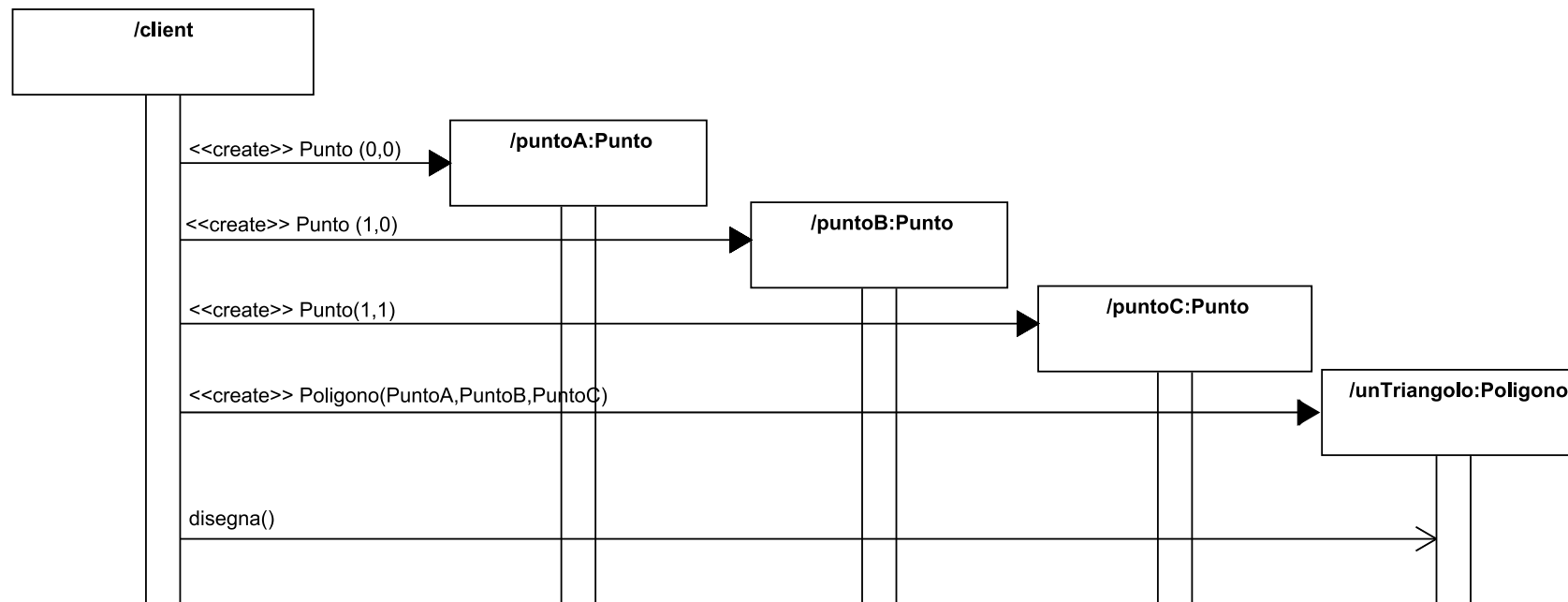
- Rappresentiamo in maniera schematica un insieme di concetti correlati tra di loro
 - non devono essere necessariamente tipi di dato di un linguaggio di programmazione (classi C++ ad es.)
 - visione ad alto livello

Esempio: class diagrams/2



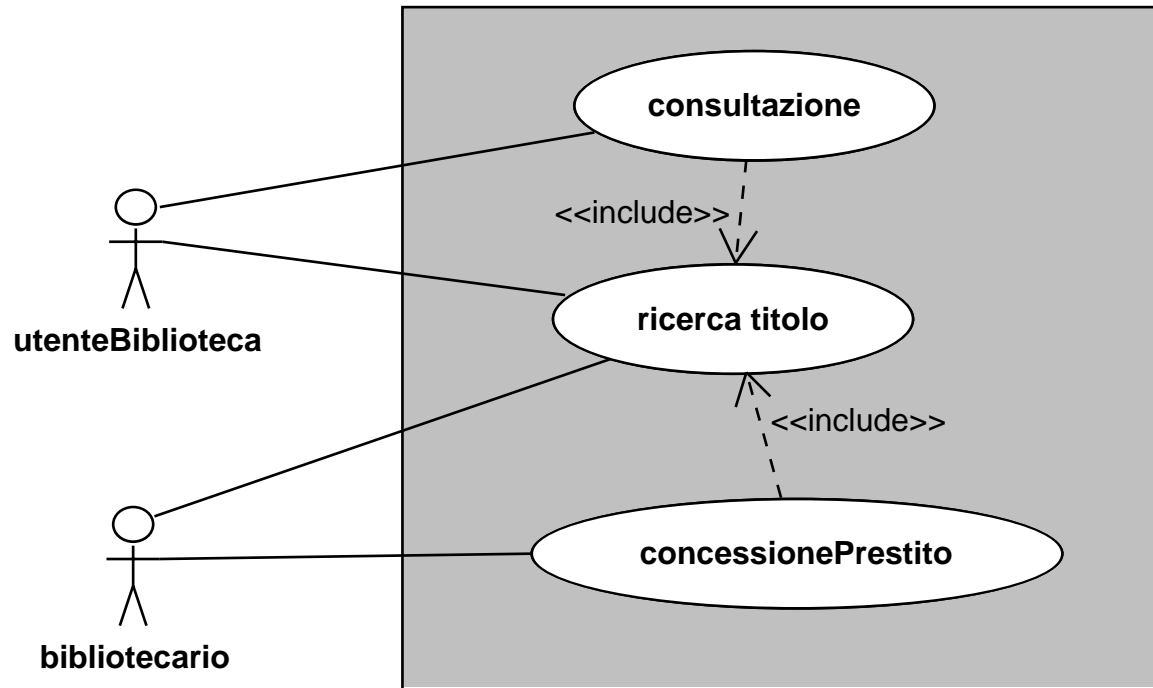
- In una fase successiva vogliamo specificare meglio l'interfaccia delle classi o parte dell'implementazione
 - nei tre compartimenti specifichiamo, oltre al nome della classe, l'insieme di attributi e l'insieme di operazioni che ciascuna classe esporta
 - una visione più dettagliata, ma il modello è lo stesso

Esempio: sequence diagrams



- Può essere utile mostrare un caso particolare di interazione tra oggetti: qui gli oggetti sono istanze delle classi descritte nei class diagrams precedenti
 - l'esempio di interazione è la costruzione di un triangolo dati i suoi punti e la richiesta disegna() all'oggetto Triangolo

Esempio: use case diagrams



- In questo caso vogliamo specificare le tipologie di utenti che possono interagire con un dato sistema e il tipo di interazione previsto per ogni tipologia
 - il punto di vista è lato utente, non ci interessa la realizzazione interna del sistema



Utilizzi di UML

- per comunicare (tra sviluppatori) un aspetto particolare del sistema che si sta progettando, in maniera immediata ma rigorosa (**sketch**)
- per fornire una descrizione completa del sistema a livello architetturale e di modulo fissando le decisioni di progetto in maniera univoca (**blueprint**)
- come un **linguaggio di programmazione** vero e proprio (servono tool specifici)

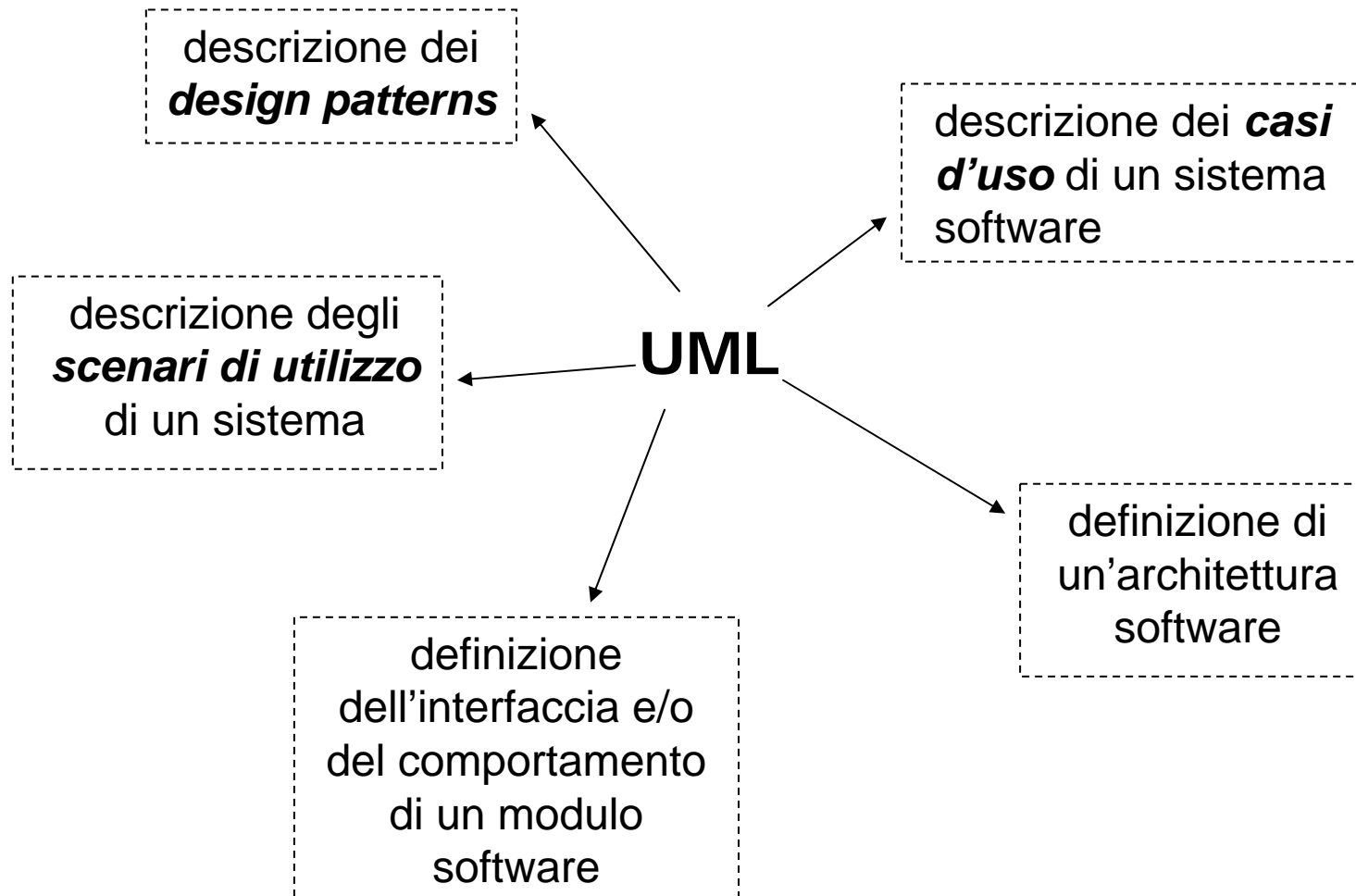
[da *UML Distilled*, Fowler]



Utilizzi di UML/2

- Analisi e specifica dei requisiti di un sistema o di parte di un sistema (ad es. use case, sequence diagrams, class diagrams)
- Progetto architeturale di un sistema (ad es. package diagrams, class diagrams)
- Definizione dell'interfaccia di un determinato modulo software (class diagrams)
- Descrizione di un algoritmo o di un procedimento di controllo (activity diagrams, state machines)
- Visualizzazione immediata di un problema ricorrente nello sviluppo software e della sua soluzione standard (class diagrams)

Come useremo/userete UML





ArgoUML*

- Uno strumento CASE per creare e gestire diagrammi UML
- open source, tecnologia Java
- **CASE**: computer aided software engineering
- non è un IDE (integrated development environment) come ad es. DevCpp, Eclipse

[*http://argouml.tigris.org/](http://argouml.tigris.org/)



ArgoUML

- permette di disegnare diagrammi UML secondo lo standard (1.4)
- permette di creare oggetti riutilizzabili in diagrammi diversi per lo stesso modello (**consistenza**) e organizzare l'intero modello in maniera automatica
- automatizza il controllo sulla **correttezza** nell'uso delle notazioni e "*consiglia*" l'utente in maniera interattiva secondo opportune linee guida
- *può essere utilizzato per generare in maniera automatica parte del codice (in C++, Java o altri linguaggi) sulla base di quanto abbiamo specificato nei diagrammi(...)*