

Unified Modeling Language

Introduzione

Sandro Morasca

Università degli Studi dell'Insubria

Dipartimento di Scienze Teoriche e Applicate

Via Ottorino Rossi 9 – Padiglione Rossi

21100 Varese, Italy

sandro.morasca@uninsubria.it



➤ Introduzione Viste

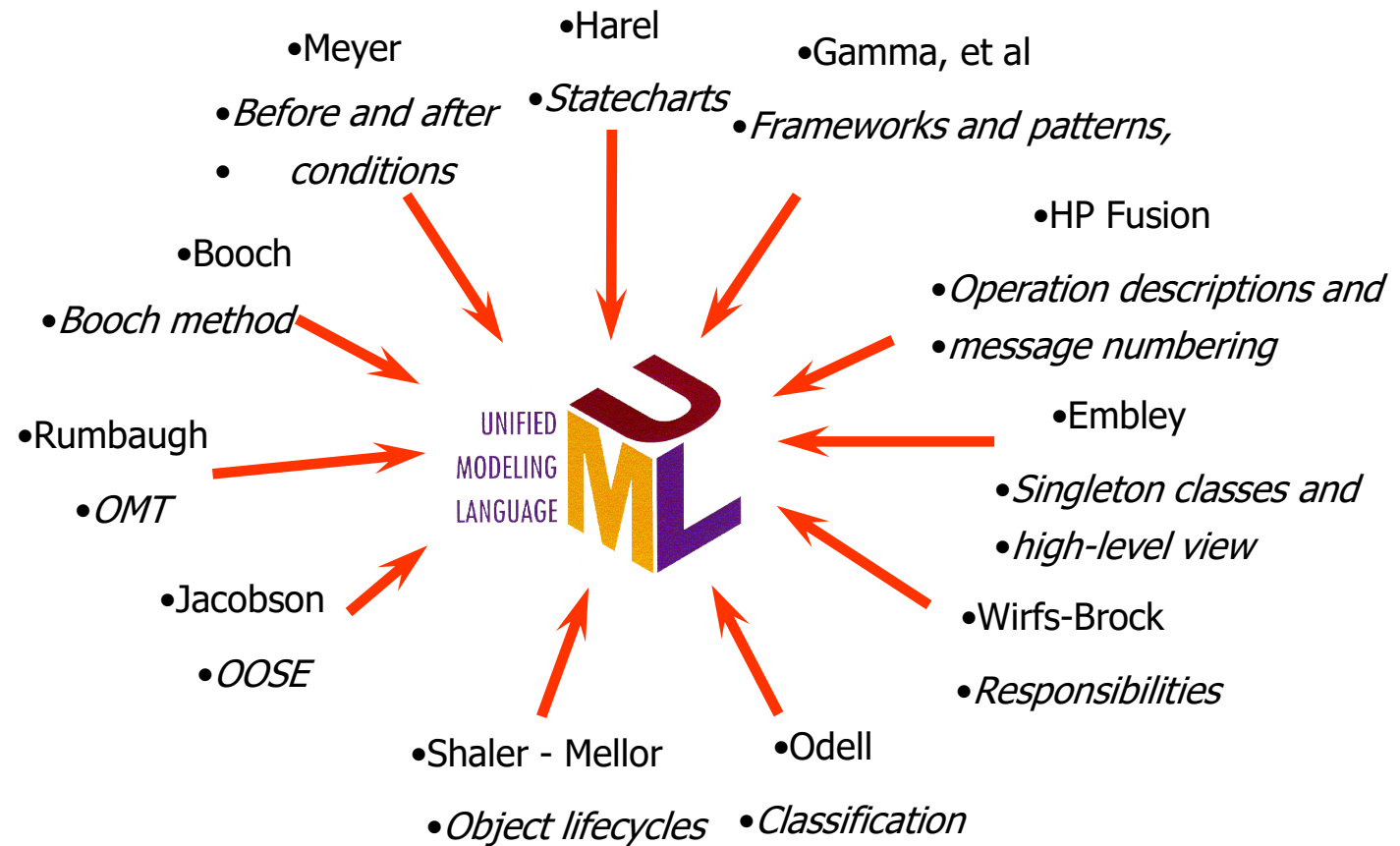
- UML è il linguaggio visuale standard per definire, progettare, realizzare e documentare i sistemi software ad oggetti
- UML riunisce molte proposte preesistenti (Booch, Rumbaugh e Jacobson)
- UML copre l'intero processo di produzione
- UML sponsorizzato dalle maggiori industrie produttrici di software
- UML è standardizzato ed è gestito dall'Object Management Group (OMG)
 - <http://www.omg.org>
 - <http://www.uml.org>



Contributi principali

UML - Introduzione

➤ Introduzione Viste





➤ Introduzione Viste

- È un insieme di linguaggi che, utilizzati congiuntamente, consentono di descrivere/modellare tutti (o quasi) gli aspetti rilevanti di un sistema, secondo un approccio object-oriented
- Notazione grafica
- Linguaggio semi-formale
- Stile misto
 - parzialmente dichiarativo, parzialmente operativo
- Standard OMG per la modellizzazione di sistemi Object-Oriented sin dal 1997



Perché UML (cont.)

UML - Introduzione

➤ Introduzione Viste

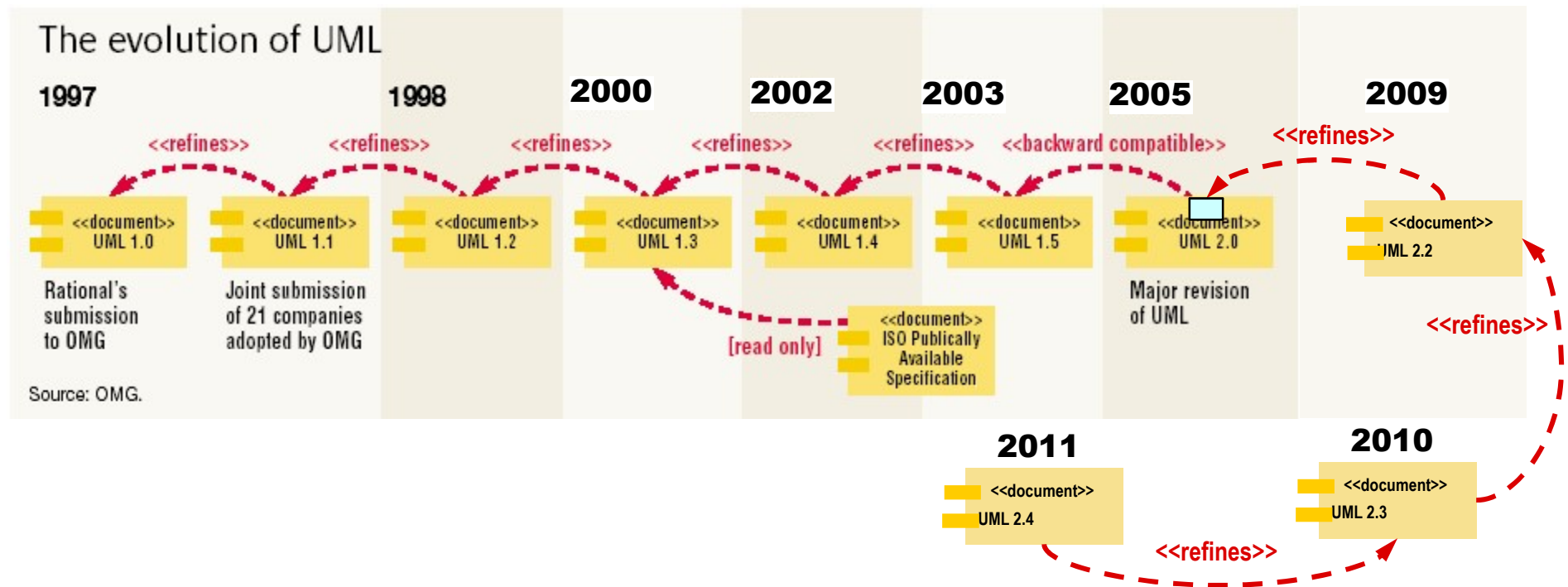
- UML riunisce aspetti dell'ingegneria del software, delle basi di dati e della progettazione di sistemi
- UML è indipendente da qualsiasi linguaggio di programmazione
- UML è utilizzabile in domini applicativi diversi e per progetti di diverse dimensioni
- UML è estendibile per modellare meglio le diverse realtà



Cronologia

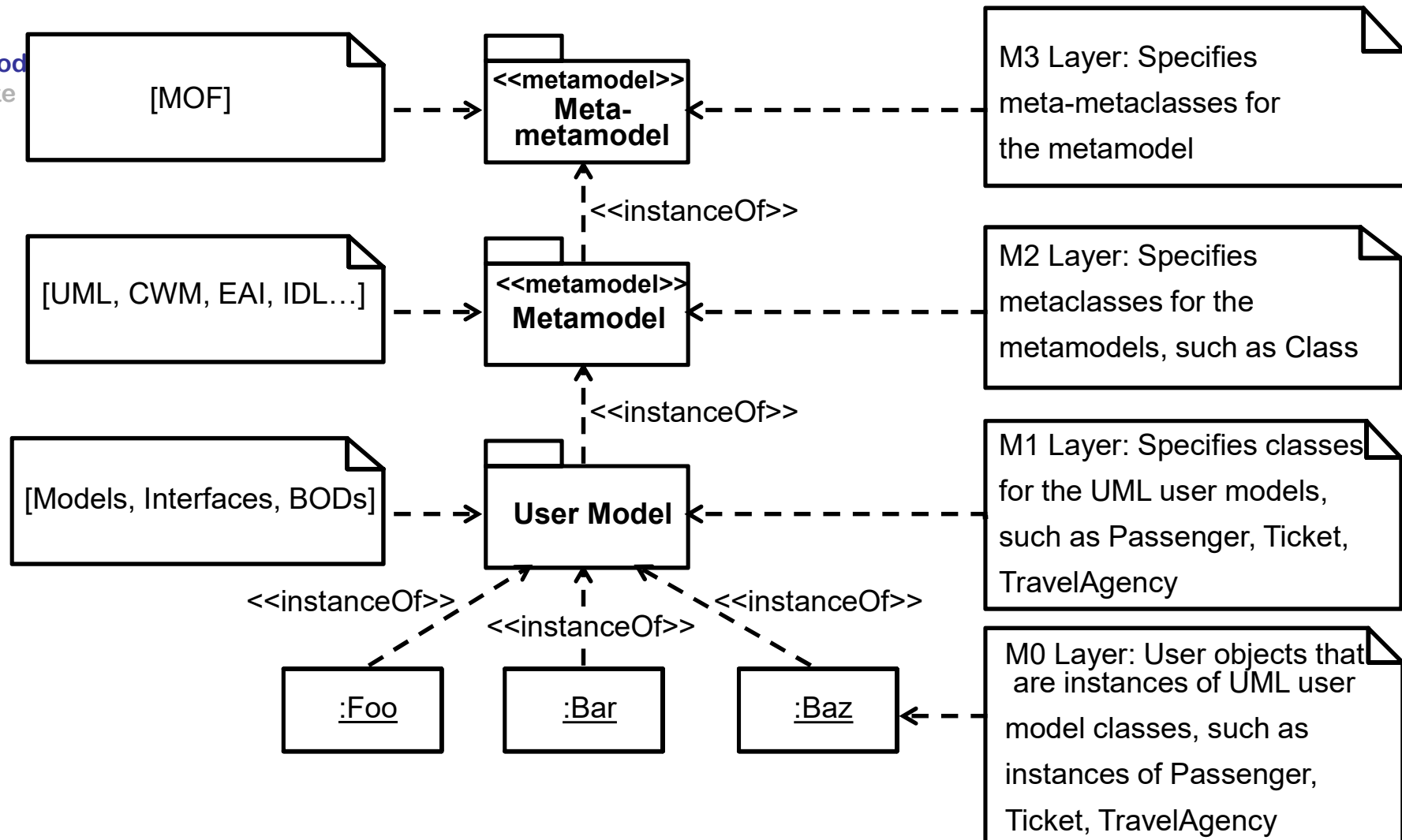
UML - Introduzione

➤ Introduzione Viste





➤ Introduzione
Viste

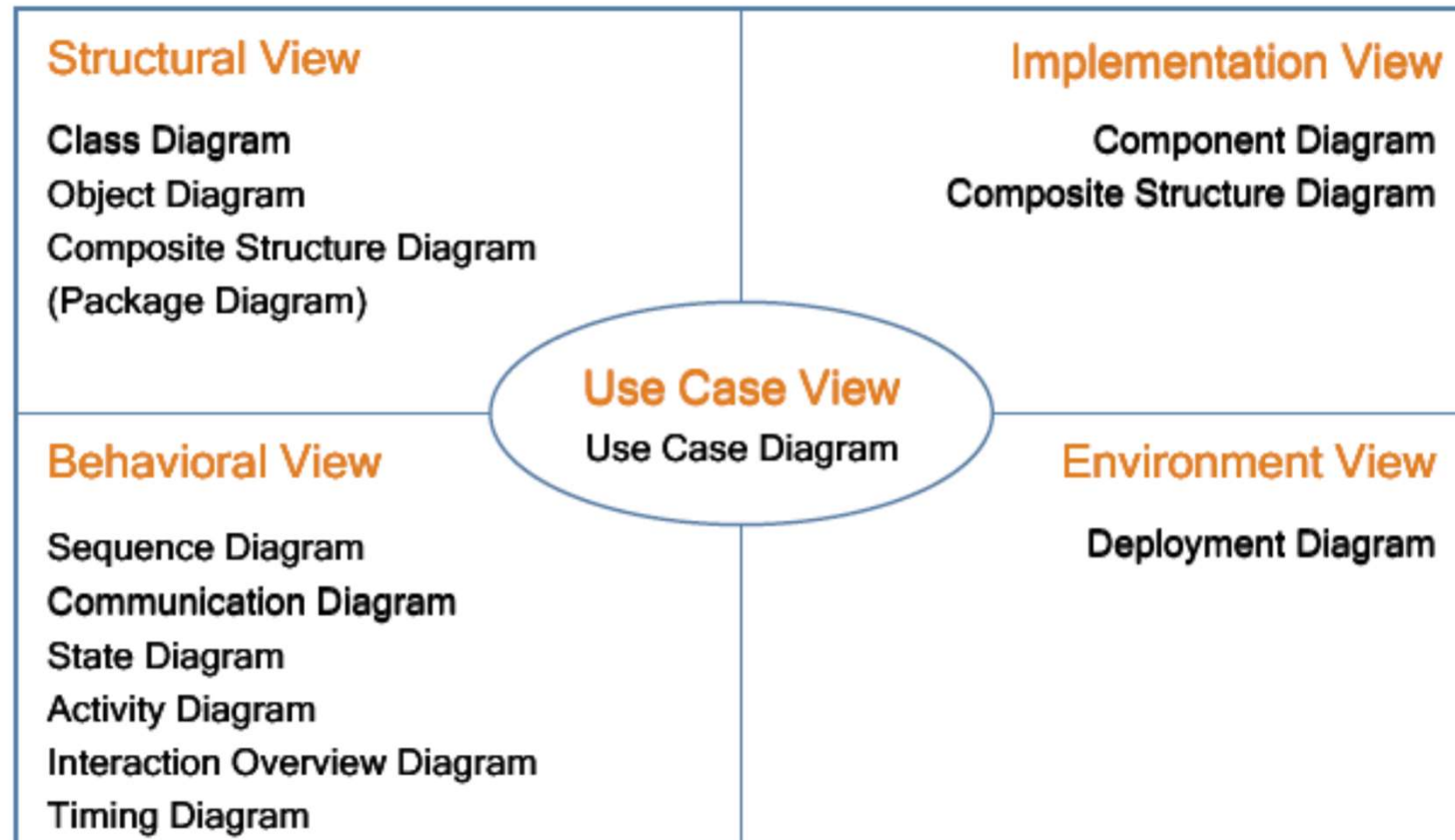




Viste di un sistema

UML - Introduzione

Introduzione ➤ Viste





- Descrive casi d'uso che forniscono valore agli utenti
- Casi d'uso essenziali vengono usati come proof of concept per l'architettura dell'implementazione
- Vengono visualizzati in UML con gli use case diagram
- Ogni use case può avere più scenari possibili
- Gli scenari possono essere descritti tramite
 - descrizioni testuali
 - graficamente tramite sequence, communication e activity diagram di UML



- Rappresenta elementi strutturali per implementare la soluzione a requisiti specifici
- Definisce
 - elementi di analisi a oggetti e di progetto
 - il vocabolario del dominio e della soluzione
 - la scomposizione del sistema in strati e sottosistemi
 - interfacce del sistema e dei suoi componenti
- È rappresentata da diagrammi statici UML
 - class diagrams con livelli di astrazione multipli
 - package diagram
 - composite structure diagram



- Rappresenta l'interazione dinamica tra i componenti del sistema per incrementare i requisiti
- Mostra la distribuzione delle responsabilità
- Permette di identificare colli di bottiglia di interazione e accoppiamento
- Può essere utilizzata per discutere requisiti nonfunzionali
 - performance, manutenibilità ...



- È specialmente importante per sistemi distribuiti
- È rappresentata da diagrammi UML dinamici
 - sequence e/o communication diagram
 - activity diagram
 - state diagram
 - interaction overview diagram
 - timing diagrams



- Descrive documenti di implementazione dei sottosistemi logici definiti nella vista strutturale
- Può comprendere documenti intermedi usati della costruzione del sistema (file di codice, librerie, file di dati ...)
- Definisce dipendenze tra componenti implementativi e le loro connessioni da interfacce richieste e fornite
- È rappresentata dai seguenti diagrammi UML
 - component diagram
 - composite structure diagram



Introduzione ➤ Viste

- Rappresenta la topologia hardware del sistema
- Definisce come vengono assegnati i componenti software ai nodi hardware
- È utile per analizzare requisiti non funzionali
 - affidabilità, scalabilità, sicurezza ...
- Fornisce informazioni per l'installazione e la configurazione del sistema
- È rappresentata da
 - deployment diagram