

INTRODUZIONE

Quick Tour

- Perché abbiamo bisogno di un modello ?
- Modellazione visuale
- Che cos'è UML ?
- Elementi base

Perché abbiamo bisogno di un modello?

- per fornire una struttura al "*problem solving*"
- per sperimentare ed esplorare più soluzioni
- per fornire l'astrazione necessaria per gestire la complessità
- per ridurre i tempi e i costi di sviluppo
- per gestire il rischio di errori

Problem solving

Generalmente il *Problem Solving* può essere definito come l'arte di risolvere problemi siano essi di natura personale, interpersonale o delle organizzazioni (aziende, enti, comunità, ecc.), o problemi software (come nel nostro caso), mediante l'utilizzo di **tattiche e tecniche**, con la massima efficacia (soluzione del problema) ed efficienza (tempo e sforzi impiegati)

Perché modellare graficamente?

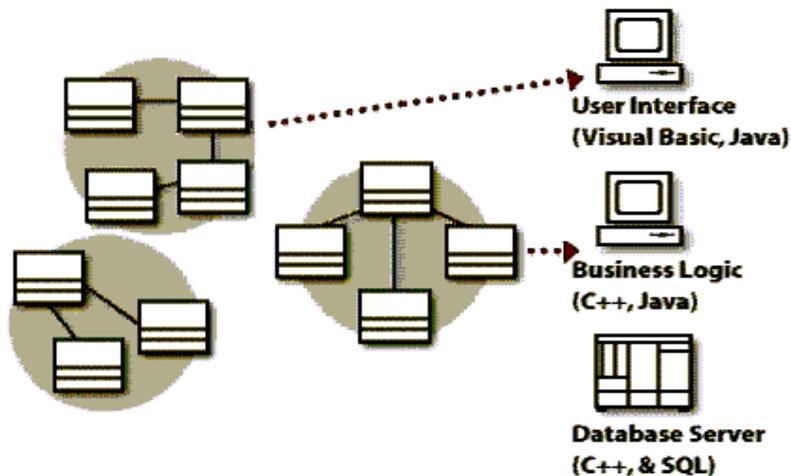
- "Graphics reveal data"
 - Edward Tufte
The Visual Display of Quantitative Information, 1983
- "1 bitmap = 1 megaword"
 - Anonymous visual modeler

Dalle slide "Object Modeling with OMG UML Tutorial Series" di Cris Kobryn

Benefici di una modellazione visuale (grafica)

- Cattura il processo aziendale (e non solo)
- Migliora la comunicazione (supera le differenze di terminologie e di linguaggi diversi)
- Gestisce la complessità (oggetti separati, viste diverse, diversi livelli di astrazione, ...)
- Definisce l'architettura
- Consente il riuso

Architettura di un sistema



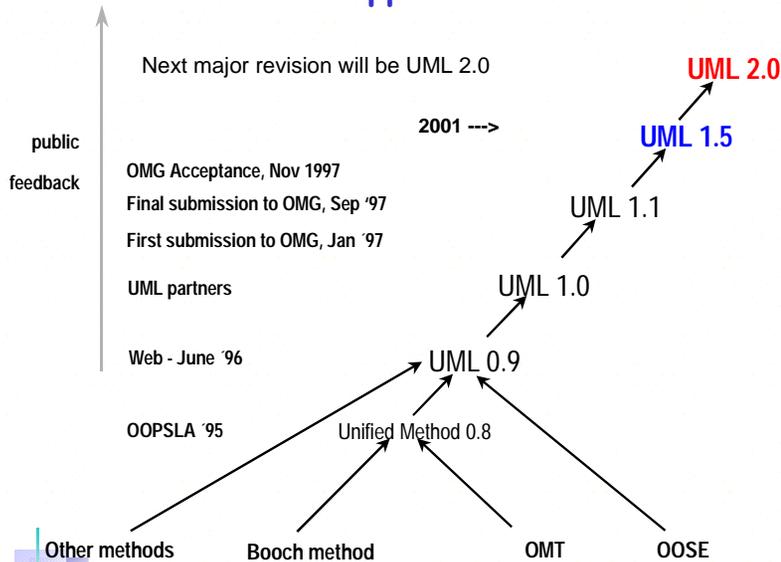
- UML è un linguaggio grafico standard per



- specificare
- visualizzare
- costruire
- documentare

gli artefatti (*artifact*) dei sistemi software

Sviluppo di UML



© Renato Conte - UML: USE CASE - 9 -

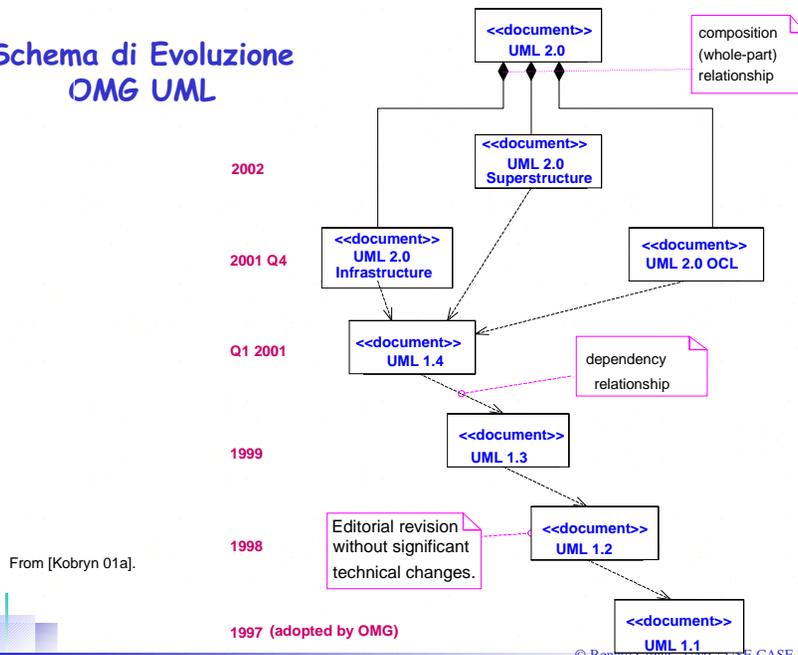
OMG

Object Management Group
organizzazione di produttori di programmi orientati agli oggetti

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

© Renato Conte - UML: USE CASE - 10 -

Schema di Evoluzione OMG UML



© Renato Conte - UML: USE CASE - 11 -

OMG UML Contributors

| | |
|---------------------------|--------------------------|
| Aonix | Microsoft |
| Colorado State University | ObjecTime |
| Computer Associates | Oracle |
| Concept Five | Ptech |
| Data Access | OAQ Technology Solutions |
| EDS | Rational Software |
| Enea Data | Reich |
| Hewlett-Packard | SAP |
| IBM | Softteam |
| I-Logix | Sterling Software |
| InLine Software | Sun |
| Intellicorp | Taskon |
| Kabira Technologies | Telelogic |
| Klasse Objecten | Unisys |
| Lockheed Martin | ... |

© Renato Conte - UML: USE CASE - 12 -

Caratteristiche di UML

- Definisce un linguaggio di modellazione visuale facile da imparare ma semanticamente ricco
- Unifica i linguaggi di modellazione di Booch, OMT, Objectory modeling language, e altri
- Incorpora la miglior esperienza sviluppata a livello industriale
- Si adatta ai moderni bisogni di sviluppo del software (scalabilità, distribuzione, concorrenza, etc.)
- Ha la flessibilità necessaria per adattarsi a diversi processi di sviluppo

Obiettivi di UML

- Fornisce meccanismi di estendibilità e specializzazione per estendere concetti base
- E' indipendente da un particolare linguaggio di programmazione o di processo di sviluppo
- Incoraggia la crescita di strumenti di sviluppo orientati agli oggetti
- Supporta concetti di sviluppo ad alto livello come: *collaborations, frameworks, patterns e components*

Il valore di UML

- E' uno standard aperto
- Supporta l'intero ciclo di sviluppo del software
- Supporta diverse "applications areas"
- E' basato sull'esperienza ed i bisogni della comunità degli utilizzatori
- Supportato da molti "tools"

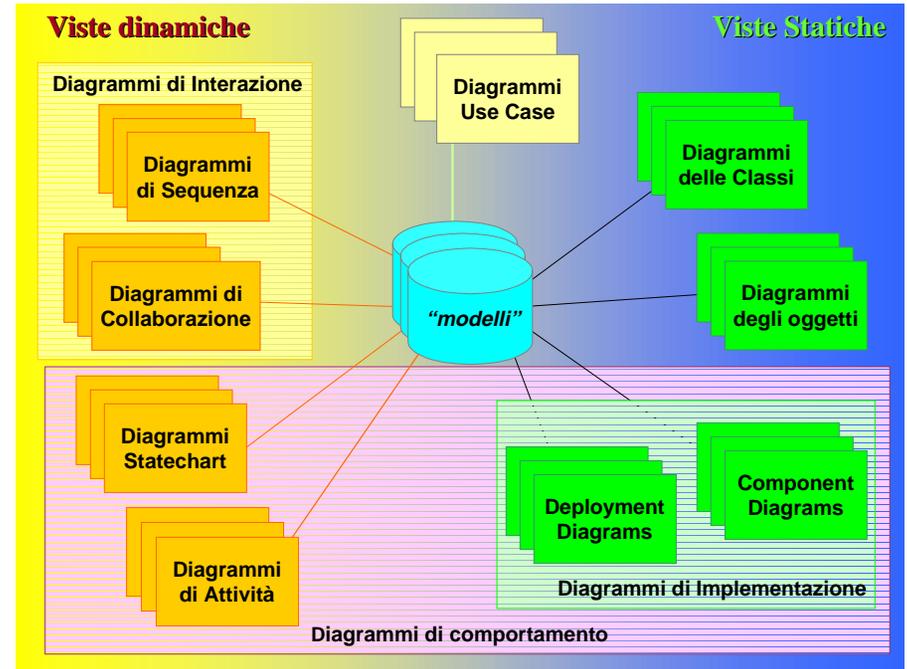
Linguaggio UML

- linguaggio = sintassi + semantica
 - sintassi = regole attraverso le quali gli elementi del linguaggio (es. parole) sono assemblate in espressioni (es. frasi, clausole)
 - semantica = regole attraverso le quali alle espressioni sintattiche viene assegnato un significato
- *UML Notation Guide*
 - definisce la sintassi grafica di UML
- *UML Semantics*
 - definisce la semantica di UML

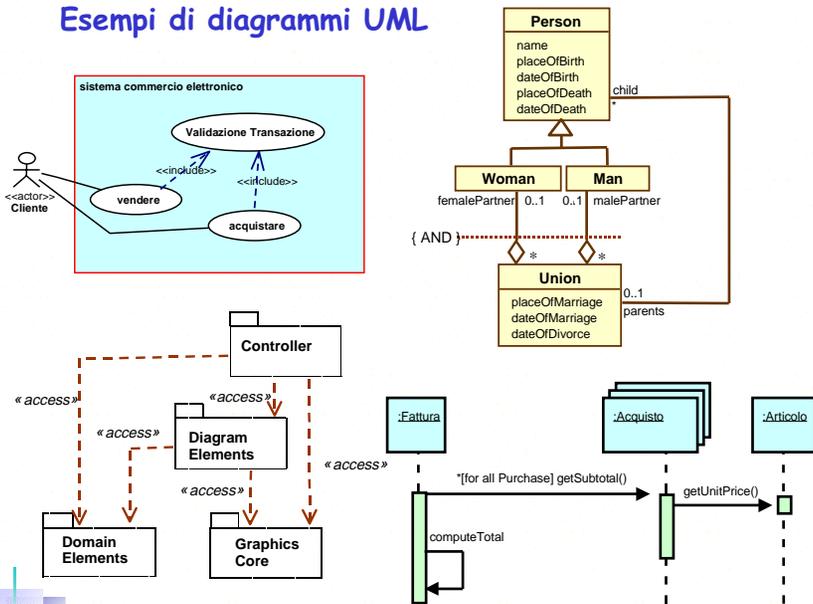
Building Blocks

- Gli elementi base di UML sono:
 - elementi di modellazione (classi, interfacce, componenti, use cases, etc.)
 - relazioni (associazioni, generalizzazioni, dipendenze, etc.)
 - diagrammi (class diagrams, use case diagrams, interaction diagrams, etc.)

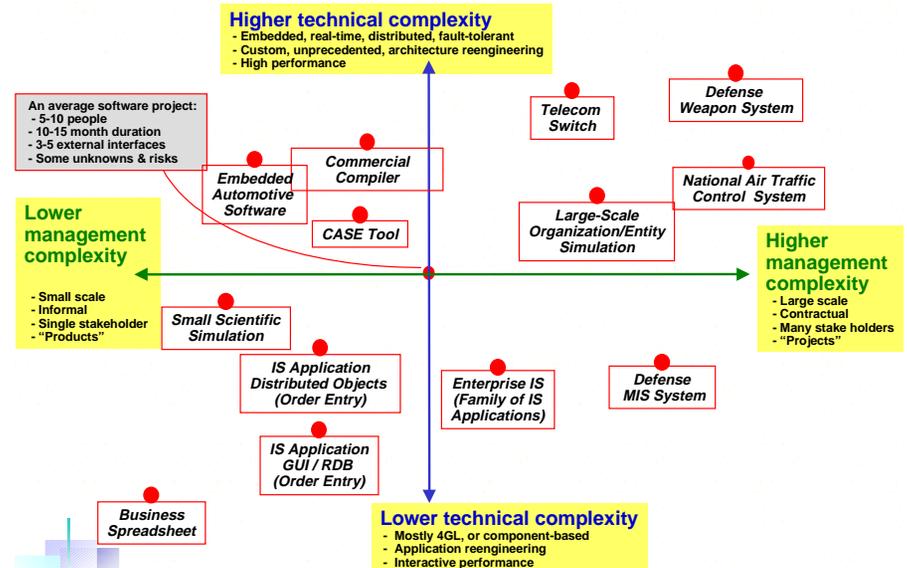
Semplici elementi base (*building blocks*) sono usati per creare strutture grandi e complesse



Esempi di diagrammi UML



Dimensioni della complessità del software



Link

- **OMG - UML resource center**
<http://www.omg.org/uml/>
- **Rational Software-UML Resource Center (IBM)**
<http://www.rational.com/uml/>
<http://www-306.ibm.com/software/rational/uml/>
- **UML - elenco di risorse e siti**
http://www.cetus-links.org/oo_uml.html

Bibliografia

The Unified Modeling Language Reference Manual

James Rumbaugh, Ivar Jacobson, Grady Booch
Addison Wesley, (1999)

The Unified Modeling Language User Guide,

Grady Booch, James Rumbaugh, Ivar Jacobson
Addison Wesley , (1999)

The Unified Software Development Process

Ivar Jacobson, Grady Booch, James Rumbaugh
Addison Wesley, (1999)