



---

Università degli Studi di Parma  
Facoltà di Scienze MM. FF. NN.  
Corso di Laurea in Informatica

Ingegneria del Software

**Il linguaggio UML**

**Giulio Destri**



Ing. SW: il ruolo di UML - 1

Giulio Destri - © AreaSP for Univ. Parma, 2006

**Scopo del modulo**

---

**Introdurre**

**L'uso di UML per l'analisi e la  
progettazione software, nel  
contesto dei sistemi informativi  
delle aziende**

Ing. SW: il ruolo di UML - 2

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

Ing. SW: il ruolo di UML - 3

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- L'inizio di ogni progetto: la modellazione
- Le specifiche utente
- Object orientation per la modellizzazione
- UML come linguaggio per descrivere la realtà, anche in vista della realizzazione di sistemi IT

Ing. SW: il ruolo di UML - 4

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Come si parte

---

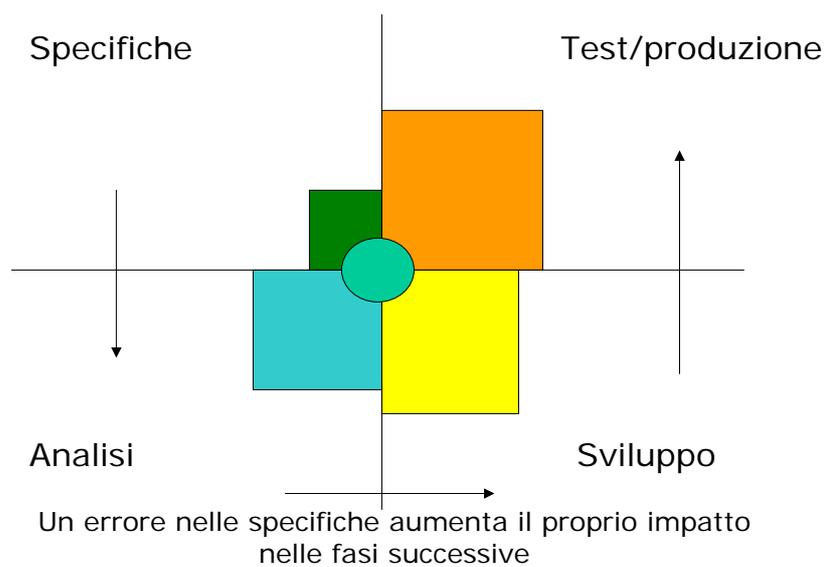
- Un progetto deve partire dalla buona conoscenza della realtà
- Si deve conoscere dove si è
- Si deve sapere dove si vuole arrivare

Ing. SW: il ruolo di UML - 5

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Impatti degli errori

---



Ing. SW: il ruolo di UML - 6

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Necessità del modeling

---

- E' necessario modellizzare la realta
- Un modello è una semplificazione della realtà che si ottiene
  - Riducendo le caratteristiche in esame
  - Considerando solo quelle utili al fine del progetto considerato/analisi in corso

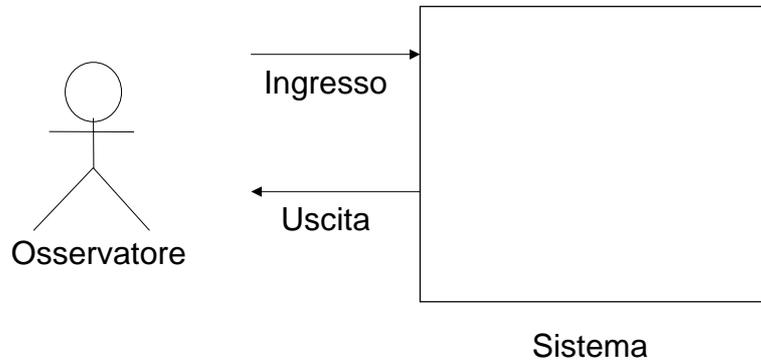
## Un sistema

---

- Un sistema è un insieme di elementi in relazione fra di loro secondo leggi ben precise, che concorrono al raggiungimento di un obiettivo comune
- Sistemi naturali
- Sistemi artificiali
- Sistemi misti

## Un sistema generico

---



Modello a “scatola nera” (black-box)

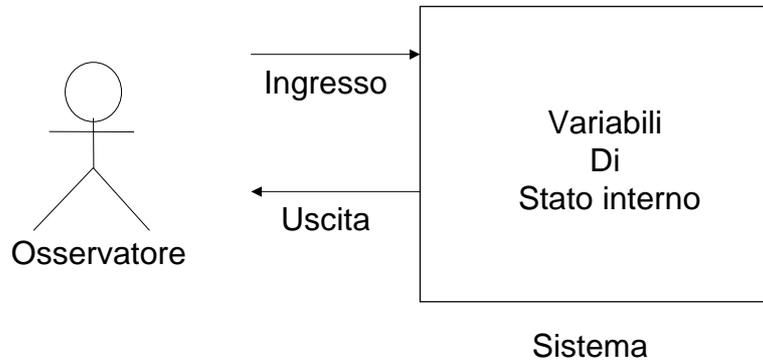
## Un modello di un sistema

---

- E' una rappresentazione del sistema stesso che, pur avendo forma e natura diverse da esso, ne conserva ed evidenzia in modo analogico alcune caratteristiche particolarmente significative per l'analisi

## Un sistema con stato interno

---



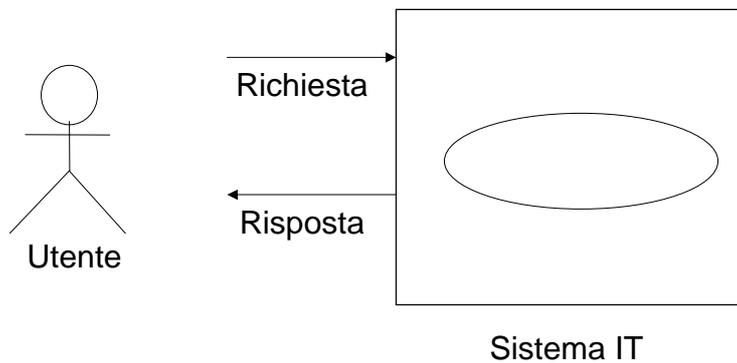
Modello con stato interno

Ing. SW: il ruolo di UML - 11

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un sistema IT generico

---



All'interno del sistema un certo processo genera la risposta alla richiesta

Ing. SW: il ruolo di UML - 12

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un processo

---

- Ogni sistema è caratterizzato da uno stato
- Il processo è la successione di stati attraverso cui un sistema passa nel corso della sua evoluzione
- Da cui deriva anche l'uso del termine processo per indicare un programma in esecuzione

## I modelli

---

- ci aiutano a "visualizzare" un sistema come è o come vorremmo che fosse
- ci permettono di specificare la struttura o il comportamento di un sistema
- ci forniscono un "template" che ci guida nella costruzione di un sistema
- documentano le decisioni che abbiamo preso

## Arrivare ai modelli

---

- Definire l'obiettivo
- Identificare il sistema e le parti interessanti
- Definire i vincoli
- Generare un modello di massima
- Formalizzare completamente il sistema
- Usare il modello (es. simulazione)

## I sistemi sono suddivisibili in componenti

---



## Per un sistema non banale

---

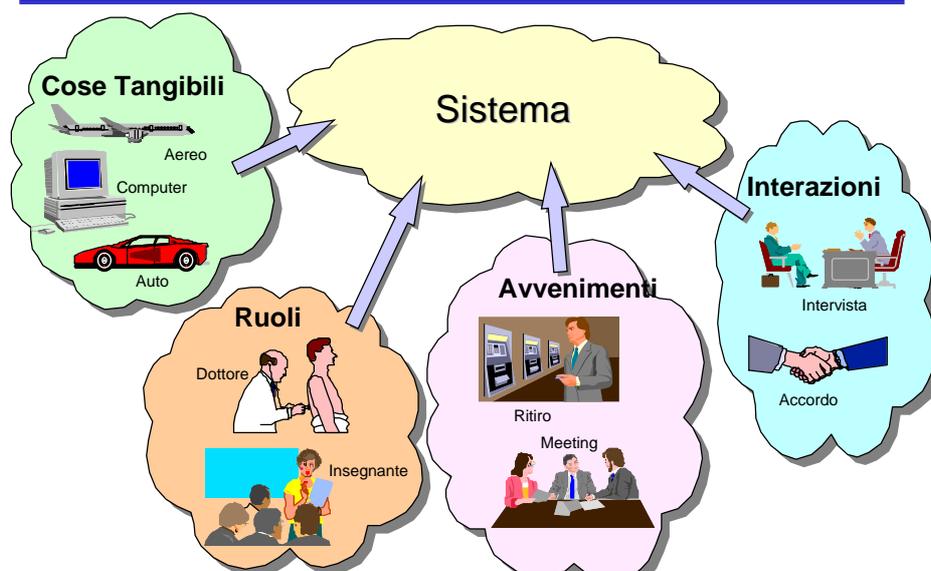
- non un solo modello
- ma un piccolo insieme di modelli,
  - che possono essere costruiti e studiati separatamente,
  - ma che sono strettamente interrelati

Ing. SW: il ruolo di UML - 17

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Identificazione dei componenti

---



Ing. SW: il ruolo di UML - 18

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Attenzione alla complessità

---

- La mente umana può elaborare 7 più o meno 2 cose contemporaneamente
- Se il modello non è ben fatto
  - Troppi dettagli – alcuni sfuggono
  - Pochi dettagli – mancanza di conoscenza

## Approcci all'analisi

---

- **Top-down:** scomposizione di un sistema per passi successivi in sottosistemi sempre più elementari
- **Bottom-up:** costruzione di un sistema complesso per composizione successiva di sistemi elementari

## Ci vuole il modello giusto

---

- Il modello deve essere valido per il contesto in cui si opera
- Deve essere adattato all'interlocutore del momento
- Vanno presi in considerazione aspetti diversi in momenti diversi e a diversi livelli di dettaglio

## Esempio: l'architetto

---

- D: "ho bisogno di una casa"
- R: "quante prese elettriche vuole in cucina?"
  
- D: "ho bisogno di una casa"
- R: "di che tipo?"
- D: "luminosa, spaziosa, sicura, funzionale, ..."

## Obiettivo: specifiche utente

---

- Lo scopo è giungere a specifiche utente chiare
- che possano essere tradotte per passi successivi
- in strumenti hardware e software

## Il riuso delle soluzioni

---

- Esempio del passare il fiume
- Soluzioni:
  - Ponte (di cemento, ferro, barche)
  - Traghetto
  - Guado
  - Caronte
  - Volare
  - A nuoto
  - ?

## OO per la modellazione

---

L'approccio Object-Oriented (OO) aiuta ad individuare

- Le componenti dei sistemi
- Il loro limite esatto
- Le azioni svolte da ciascuna componente
- Le relazioni fra le varie componenti

## Gli strumenti di modellazione

---

- Sono (e devono rimanere) un ausilio per la modellazione
- Devono essere finalizzati ad una maggiore comprensione di un sistema
- Non devono dare adito a metodologie rigide ma ad una maggiore flessibilità

## **Modellazione: scomporre un sistema**

---

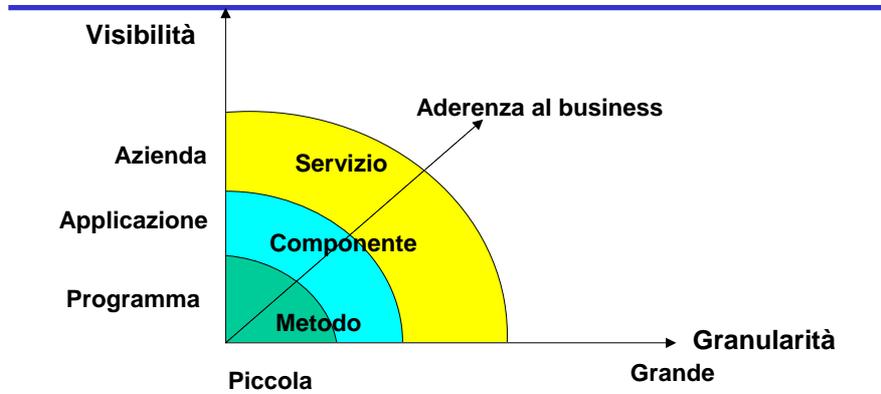
- Non esiste un approccio generale, ma lo scopo per cui stiamo lavorando ci deve guidare
- La granularità della scomposizione dipende dal problema e da dove si vuole arrivare

## **Granularità per la scomposizione di un sistema**

---

- Dati entità, sottosistemi, sistemi...
- Una singola entità, osservata da un certo punto di vista, entro il contesto di un problema, può essere un intero sistema quando è collocato entro un altro

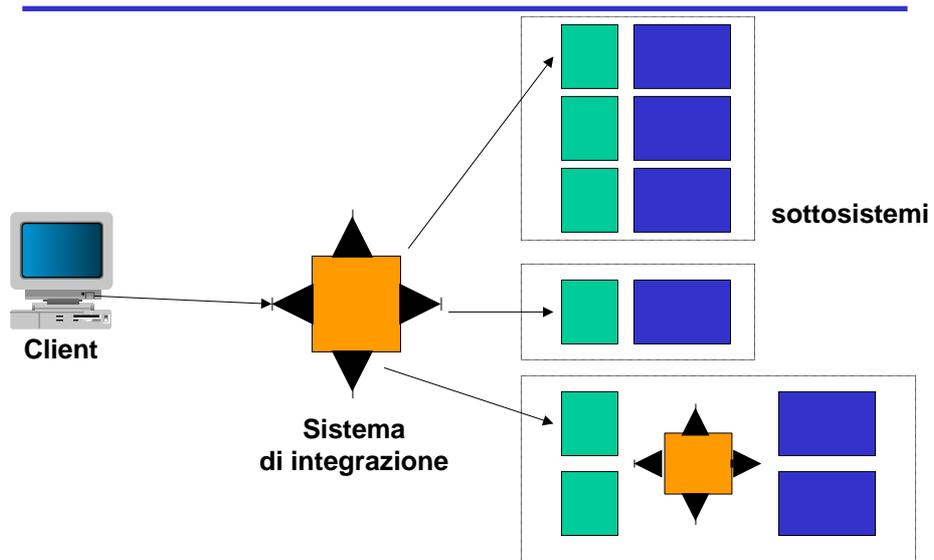
## Granularità per la scomposizione di un sistema - 2



Ing. SW: il ruolo di UML - 29

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Granularità per la scomposizione di un sistema - 3



Ing. SW: il ruolo di UML - 30

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

## Il sistema informativo: definizione

---

“L'insieme di persone, apparecchiature, procedure aziendali il cui compito è quello di produrre le informazioni che servono per operare nell'impresa e gestirla”.

(M. De Marco)

Corrisponde all'inglese “Information System”

## I sistemi informativi: composizione

---

Pertanto un sistema informativo si suddivide in:

- **Risorse umane** (con organizzazione, ruoli, esperienze, ecc...)
- **Risorse tecnologiche** (sistema informatico, inglese "IT System")
- **Risorse organizzative** (procedure, regolamenti, workflow, ecc...)

## Un sistema informativo è un sistema

---

Anche il sistema informativo

- è un insieme di elementi
- in relazione fra di loro
- secondo leggi ben precise
  
- che concorrono (quasi sempre) al raggiungimento di un obiettivo comune

## Un sistema informativo è un sistema (2)

---

Pertanto

- non è corretto considerare solo gli aspetti tecnologici di un sistema informativo
- ma va considerato nel suo insieme

## Sistemi informativi e tecnologia

---

- La risorsa tecnologica è comunque un componente fondamentale per tutto il sistema informativo
- Sono possibili diversi "pattern di interazione" per il rapporto tra il sistema informativo e la tecnologia, e la conseguente evoluzione del primo

## Sistemi informativi: evoluzione

---

- **Technological imperative:** una nuova disponibilità IT "impone" il cambiamento
- **Organizational imperative:** nuove necessità organizzative impongono il cambiamento
- **Emergent perspective:** l'interazione con una nuova tecnologia conduce al cambiamento

## Il sistema informatico: suddivisione

---

- Informatica utente (postazioni client, suite Office, e-mail, applicativi client)
- Applicativi aziendali (ERP, business intelligence, supporti progetto ecc...)
- Basi di dati (DB relazionali, archivi destrutturati ecc...)

## Il sistema informatico: suddivisione (2)

---

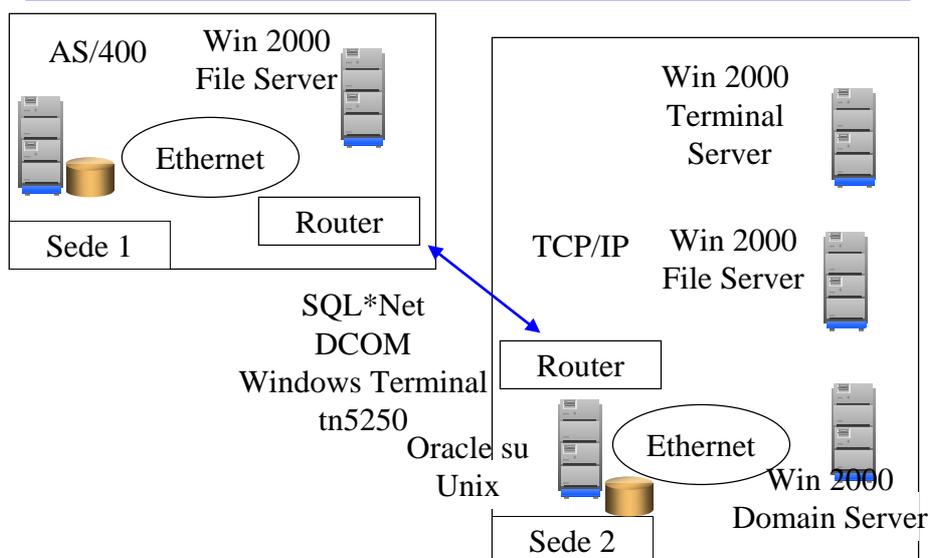
- Sistemi gestione accessi (es. LDAP, RACF ...)
- Sistemi operativi
- Infrastrutture di rete

Ing. SW: il ruolo di UML - 39

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un esempio di sistema informatico

---



Ing. SW: il ruolo di UML - 40

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Il sistema informatico: evoluzione

---

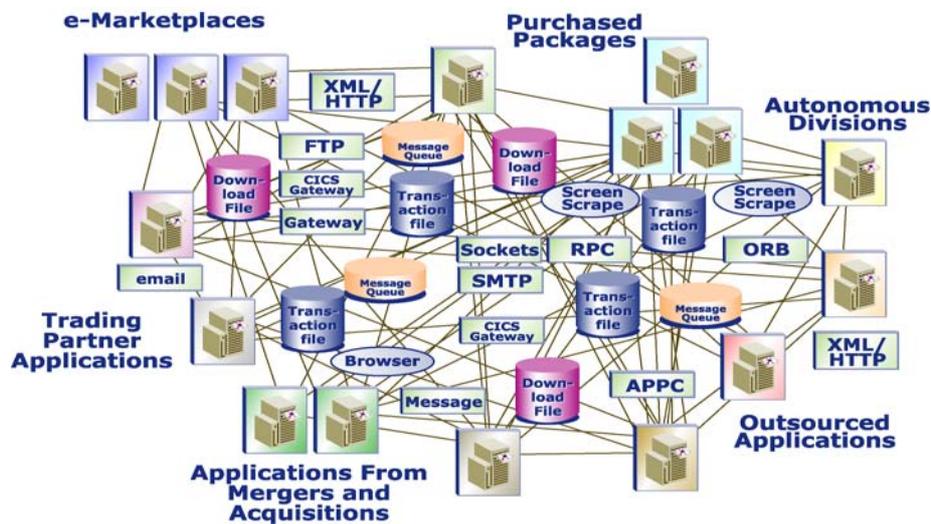
- Spesso nuovi applicativi
- realizzati per rispondere alle mutate esigenze del business aziendale
- devono integrarsi con applicazioni ancora efficienti
- la cui architettura è però ormai datata

## Il sistema informatico: evoluzione

---

- si ha quindi la convivenza di applicazioni
  - realizzate in epoche differenti
  - su piattaforme molto eterogenee
- Che devono collaborare (e quindi comunicare fra loro)

## Problema: spaghetti-integration



Ing. SW: il ruolo di UML - 43

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

Ing. SW: il ruolo di UML - 44

Giulio Destri - © AreaSP for Univ. Parma, 2006

## UML

---

- Unified Modeling Language
- Linguaggio unificato per la modellazione di concetti, entità, funzionalità, processi e relazioni che fra essi intercorrono
- Nasce nel 1997
- Unisce precedenti sintassi di modellazione (Booch, OMT, OOSE)
- Standard gestito dal consorzio OMG (<http://www.omg.org>)

## UML come strumento di modellazione

---

Il linguaggio UML serve a descrivere, in modo grafico e "compatto"

- I requisiti utente (use case)
- Le componenti dei sistemi
- I dati in esse contenuti
- Le azioni da esse svolte
- Le relazioni che fra loro intercorrono (il processo in cui esse operano)

## Evoluzione di UML

---

- Dopo il 1997 le estensioni standard di UML si susseguono
- Anche l'uso viene esteso alle varie fasi di realizzazione dei sistemi IT
- Attualmente il riferimento è lo standard 1.5 (molti strumenti usano ancora l'1.3)
- E' appena stato rilasciato lo standard 2.0 e già qualche strumento lo implementa

## Obiettivo di UML

---

- Linguaggio di modellazione, semigrafico, utilizzabile da tecnici, da non tecnici e dalle macchine
- Un modello è una semplificazione della realtà che si ottiene
  - Riducendo le caratteristiche in esame
  - Considerando solo quelle utili al fine dell'analisi in corso
- Potenza espressiva nella documentazione

## L'evoluzione logica di UML

---

- Nella catena di produzione dei sistemi IT, l'accento viene posto sempre più sulle fasi di analisi e progettazione
- Una fase di analisi deve provvedere la conoscenza di dove si è (AS IS) e dove si vuole andare (TO BE)
- Uso di UML anche nel Business Modeling

## Argomenti

---

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- [Dall'IT modeling al business modeling](#)
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

## Uso di UML: IT

---

- Interazione con l'utente
  - Per rendere precise le descrizioni (Use Cases)
  - Per aiutare l'utente a chiarire le proprie necessità
- Ausilio, guida e documentazione nelle fasi di sviluppo
- Memoria storica dello sviluppo e del sistema realizzato

## Uso di UML: IT e Business Modeling

---

- Il sistema informativo è l'insieme delle risorse umane, materiali, organizzative (regole) che gestisce e distribuisce le informazioni entro l'azienda
- Il sistema informatico è l'insieme degli strumenti IT di un sistema informativo
- Serve un linguaggio chiaro comune a tutti i componenti del sistema informativo

## Uso di UML: Business Modeling

---

- Usare le stesse notazioni di UML per
  - Esprimere i concetti del business
  - Per uso diretto (per uso di analisi business)
  - Per uso indiretto (in vista di una successiva implementazione di sistemi IT di ausilio al business)

## Uso di UML: Business Modeling

---

- Usare UML in tutto il sistema informativo, come linguaggio comune
- Il processo di sviluppo software usa lo stesso linguaggio sin dalle prime fasi dell'analisi
- Comunicazione con fornitori IT con linguaggio comune

## Uso di UML: Business Modeling

---

- Metodologie unificate
  - RUP (Rational Unified Process) di Rational (<http://www.rational.com>)
  - Scuola scandinava (Eriksson e Penker) (<http://www.opentraining.com>)
- Sistemi IT "full-object"
  - Zucchetti

## Argomenti

---

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- [Come si usano i diagrammi](#)
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

## Tipi di diagrammi UML: analisi

---

- dei casi d'uso - **Use Case Diagram**
- delle classi - **Class Diagram**
- degli oggetti – **Object Diagram**
- di sequenza - **Sequence Diagram**
- di collaborazione - **Collaboration Diagram**
- di transizione di stato - **Statechart Diagram**
- delle attività - **Activity Diagram**

Ing. SW: il ruolo di UML - 57

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di diagrammi UML: implementazione

---

- dei componenti - **Component Diagram**
- di distribuzione - **Deployment Diagram**

Ing. SW: il ruolo di UML - 58

Giulio Destri - © AreaSP for Univ. Parma, 2006

## L'uso dei diagrammi UML: analisi

---

- **Use Case Diagram:** per capire nei dettagli "cosa" il sistema deve fare
- **Class/Object Diagram:** per definire le entità fondamentali
- **Activity/Statechart Diagram:** per definire i processi fondamentali (fortemente "imparentati" con gli Use Case Diagram)
- **Collaboration Diagram:** per definire le interazioni fra le entità fondamentali
- **Sequence Diagram:** per definire la sequenza delle interazioni fra entità

## L'uso dei diagrammi UML: implementazione

---

- **Component Diagram:** per definire nei dettagli quali parti compongono il sistema/prodotto software finito e le loro relazioni
- **Deployment Diagram:** per definire dove le varie parti di un programma devono essere poste in un'architettura distribuita (es. client-server)

- **Use Case Diagram**
- **Class Diagram**
- **Sequence Diagram**
- **Collaboration Diagram**
- **Activity Diagram**
- **Statechart Diagram**

## Lo Use Case (caso d'uso)

---

- Lo use case è un contratto
- Descrive l'interazione fra due entità che interagiscono fra loro
- Consente di stabilire :
  - **Servizi forniti**
  - **Servizi richiesti**
  - **Utenti abilitati**
  - **Vincoli nell'erogazione**

## Lo Use Case (caso d'uso)

---

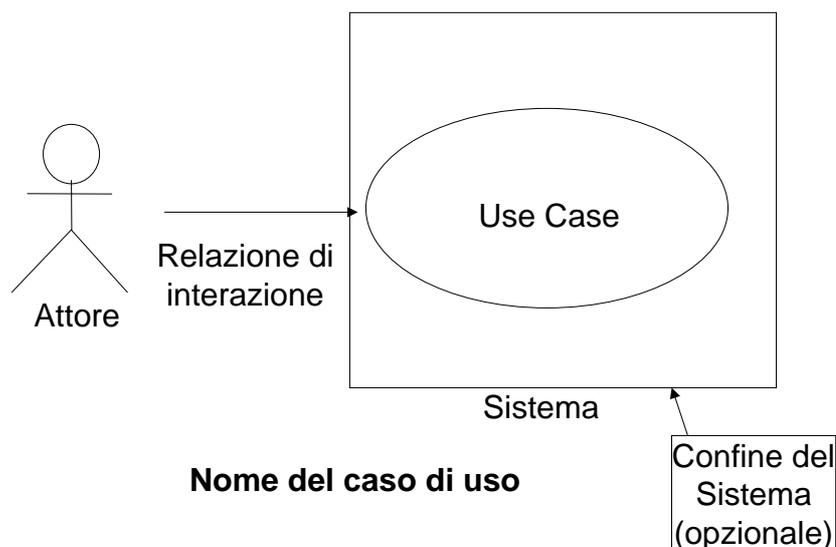
- Il primo scopo è quello di trovare un confine preciso (boundary) per il sistema/sottosistema/componente che si sta analizzando
- Una volta definito il confine si può stabilire
- **cosa fa il sistema rispetto all'esterno**
- **e identificare attori e use case**

Ing. SW: il ruolo di UML - 63

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Use Case Diagram: sintassi

---



Ing. SW: il ruolo di UML - 64

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Definizione formale di uno Use Case

---

SEQUENZA DI TRANSAZIONI,  
eseguita da un ATTORE  
in interazione col SISTEMA,  
la quale fornisce  
un VALORE MISURABILE  
per l'attore

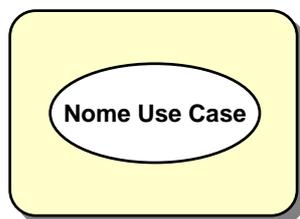
Definisce le richieste al sistema  
dipendenti dall'attore

Ing. SW: il ruolo di UML - 65

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Definizione pratica di Use Case

---



- Sequenza di transazioni in dialogo col sistema
- Comporta Sempre uno o più Attori
- Rappresenta COSA (non come) il sistema offre all'attore
- Mappato alle attività di business

Gli Use case definiscono fabbisogni di sistema  
"testabili" con una vista "fuori-dentro"

Ing. SW: il ruolo di UML - 66

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Cos'è un attore

---

- Un attore rappresenta un'entità esterna al sistema (una persona, un altro sistema software, un componente hardware) che interagisce col sistema
- Un attore individua un ruolo piuttosto che un'entità fisica

## Cos'è uno Use Case (caso d'uso)

---

- Uno use case rappresenta una situazione tipica di utilizzo del sistema e comprende in sé vari flussi possibili di esecuzione
- Uno use case rappresenta un'importante parte di funzionalità, completa dall'inizio alla fine

## Come si trova un attore?

---

- Un attore è un **ruolo**
- Usare le domande:
  - chi ha bisogno del sistema? (il caro vecchio: cui prodest?)
  - chi userà le funzionalità principali?
  - chi dovrà mantenere e amministrare il sistema?
  - di quali dispositivi hardware il sistema ha bisogno?
  - con quali altri sistemi il sistema dovrà comunicare?

## Come si trova uno use case?

---

Per ognuno degli attori precedentemente identificati, si può rispondere alle seguenti domande:

- quali funzioni l'attore richiede al sistema?
- l'attore ha bisogno di leggere o scrivere o immagazzinare informazioni nel sistema?
- l'attore deve ricevere notifiche di eventi dal sistema?

## Attori e Use Case

---

- Attori e use case sono sempre collegati fra loro
- un attore isolato non può interagire col sistema
- uno use case isolato non fornisce alcuna funzionalità all'esterno (stiamo parlando di funzionalità che abbiano un senso all'esterno)

## Attori e Use Case - 2

---

Un attore può essere

- **attivo**, ovvero inizia uno use case
- **passivo**, ovvero partecipa a uno use case, ma non lo inizia

## Perché usare gli Use Case?

---

PER EVITARE :

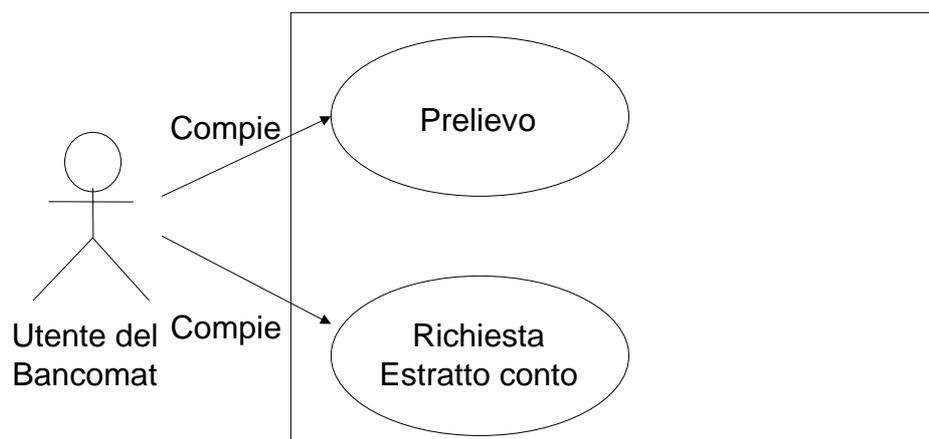
- *Utente*: non è ciò che volevo !
- *Analista*: ma come, fa tutto quello che mi hai chiesto...
- *Utente*: sì, ma non come intendevo !

Ing. SW: il ruolo di UML - 73

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Use Case Diagram: esempio

---

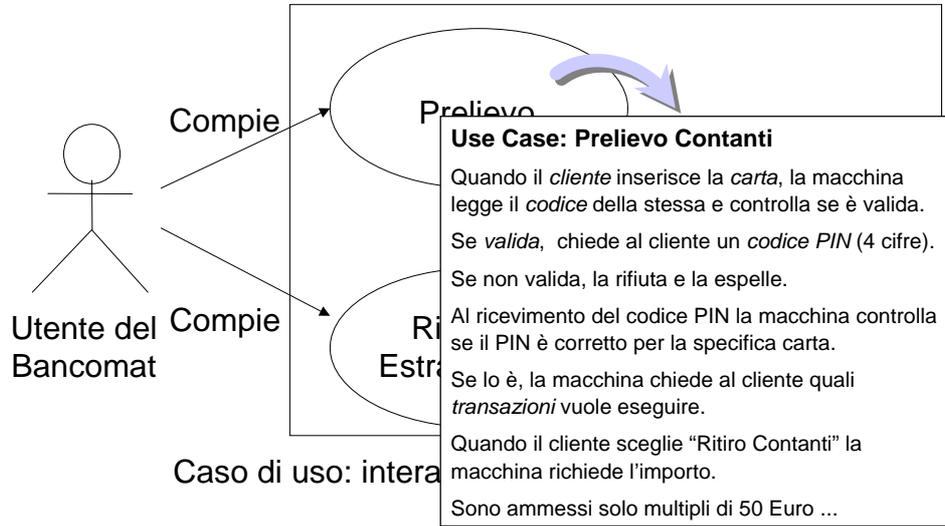


Caso di uso: interazioni con il bancomat

Ing. SW: il ruolo di UML - 74

Giulio Destri - © AreaSP for Univ. Parma, 2006

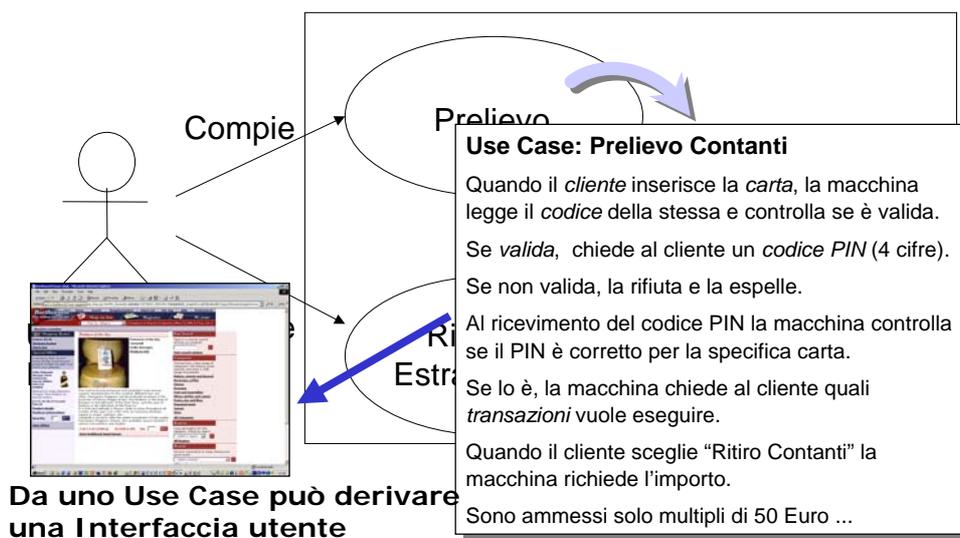
## Descrizioni Use Case: Specificano i Dettagli



Ing. SW: il ruolo di UML - 75

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Use Case Diagram e Interfacce Utente

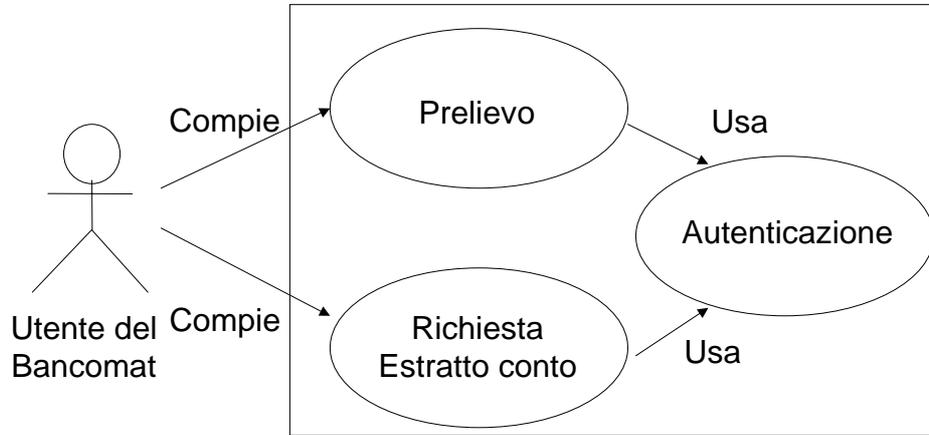


Da uno Use Case può derivare  
una Interfaccia utente

Ing. SW: il ruolo di UML - 76

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Use Case Diagram: scomposizione

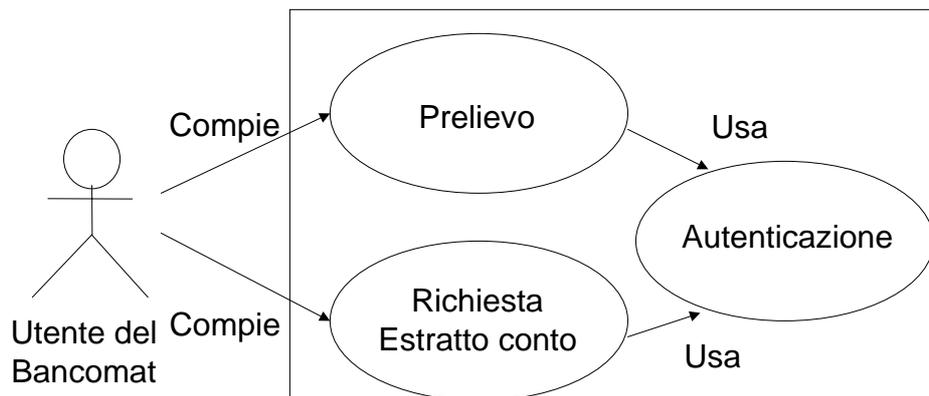


Caso di uso: interazioni con il bancomat

Ing. SW: il ruolo di UML - 77

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Modello dei rapporti tra Attori e Use Case



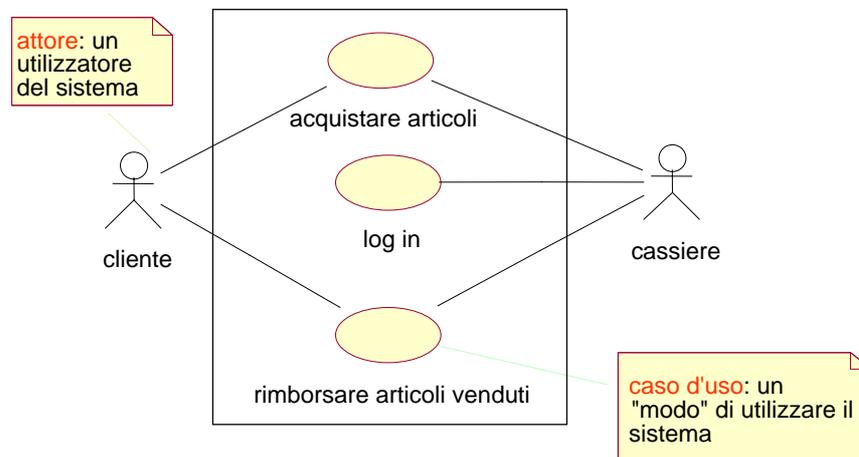
- ◆ Attori rappresentati da una persona
- ◆ Use case = ellissi
- ◆ Astrazione delle interazioni tra Attori ed uno use case
- ◆ Il sistema è circondato dal riquadro

Ing. SW: il ruolo di UML - 78

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio di Use Case: la cassa

---



Ing. SW: il ruolo di UML - 79

Giulio Destri - © AreaSP for Univ. Parma, 2006



- Use Case Diagram
- **Class Diagram**
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Statechart Diagram

Ing. SW: il ruolo di UML - 80

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Gli oggetti

---

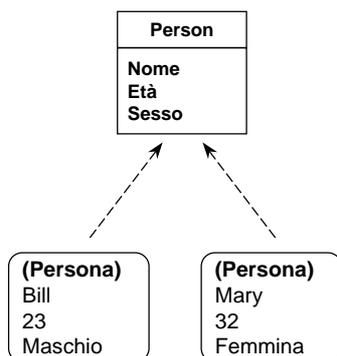
- Gli oggetti sono proiezioni delle entità del mondo reale (del dominio del problema) entro il dominio dell'applicazione software
- Possono essere usati anche come strumenti di modellizzazione per l'approccio OO

Ing. SW: il ruolo di UML - 81

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Gli Oggetti sono Istanze delle Classi

---



Classificazione: aspetto chiave dell'Object-Oriented

- **Classe ...**
  - Astrae le proprietà essenziali degli oggetti
  - Definisce gruppi di oggetti con proprietà, comportamento, interazioni e semantica simili
  - Fornisce un "modello" per costruire un oggetto (è una "fabbrica delle istanze")
  - Definisce le operazioni fornite dalle istanze
- **Oggetti ...**
  - Sono definiti dalla loro classe
  - Hanno un valore intrinseco per ogni proprietà definita dalla loro classe
  - "Conoscono" la loro classe
  - Possono essere creati "al run time"

Ing. SW: il ruolo di UML - 82

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le classi

---

- Una classe rappresenta una entità di business coinvolta nell'elaborazione (class diagram di business)
- L'individuazione delle classi consente di identificare chi fa che cosa e come all'interno del sistema/azienda
- La struttura delle classi aziendali relative ad un processo può essere confrontata con la definizione di strutture efficienti per la realizzazione (pattern)

## Le classi

---

- Una classe può rappresentare un oggetto concreto (un tornio) oppure immateriale (Iva) o una attività (spedizione)
- Una classe ha delle caratteristiche che la descrivono (attributi), ha delle elaborazioni che vengono eseguite (operazioni), rappresenta una definizione applicabile ad un insieme di elementi omogenei (oggetti)

## Le classi

---

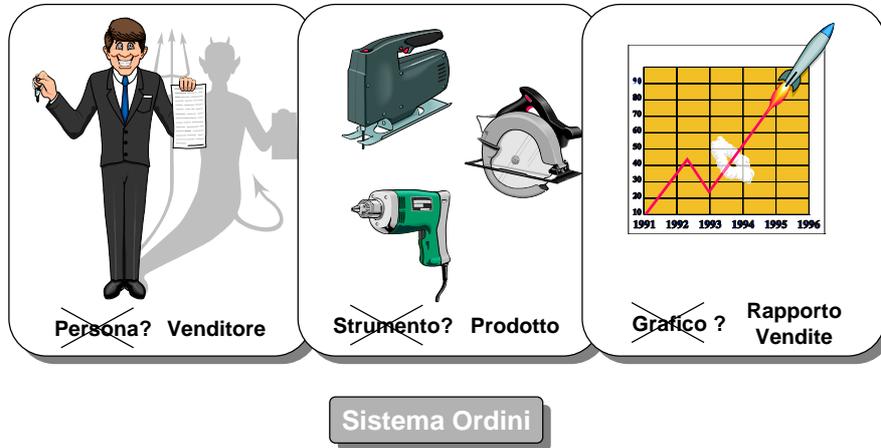
- Le classi sono proiezioni delle entità del dominio del problema entro il dominio del modello
  - Una classe definisce gruppi di entità con proprietà, comportamento, interazioni e semantica simili
  - Una classe fornisce un "modello" per "costruire" ciascuna di tali entità

## Le classi

---

- Classe
  - Prototipo
  - Modello
  - Astrazione
  - Definisce Proprietà e comportamento degli oggetti

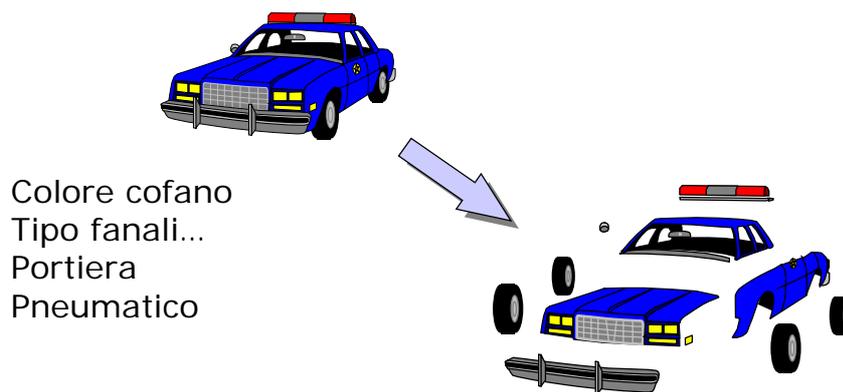
## Le Classi devono avere Nomi Concisi e Precisi, presi dal linguaggio del dominio



Ing. SW: il ruolo di UML - 87

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le classi hanno degli attributi

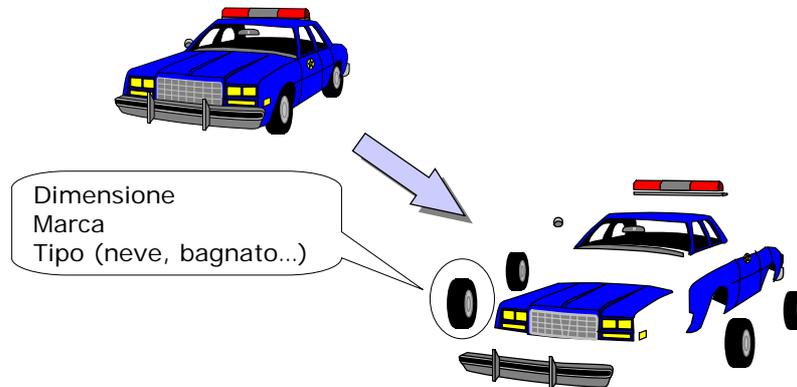


Ing. SW: il ruolo di UML - 88

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le classi possono avere componenti

---



Ing. SW: il ruolo di UML - 89

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Oggetti e classi

---



- Classe ...

- Analoga ad una definizione presente in un dizionario o un'enciclopedia
- Esempio: l'auto della polizia



- Oggetti ...

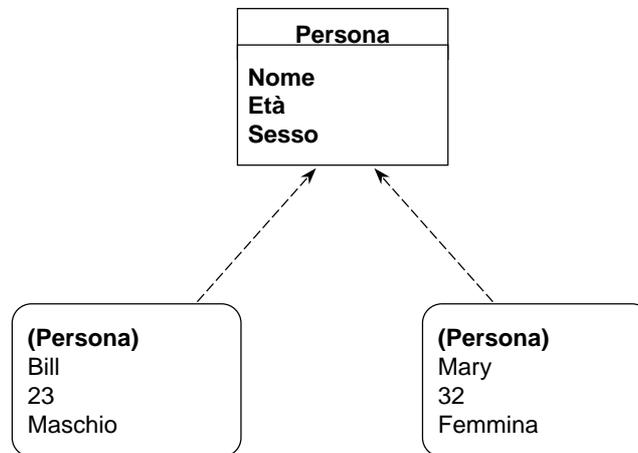
- Oggetto concreto
- Esempio: l'automobile n.12 della polizia municipale di Parma

Ing. SW: il ruolo di UML - 90

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Gli Oggetti sono la "concretizzazione" delle Classi

---



Ing. SW: il ruolo di UML - 91

Giulio Destri - © AreaSP for Univ. Parma, 2006

## I class diagram

---

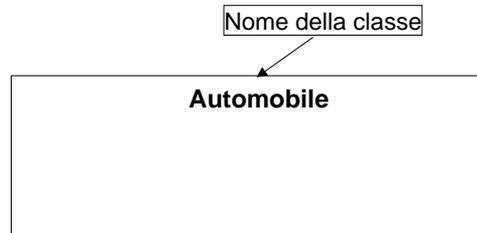
- Modellano la relazione fra le entità del sistema, rappresentate come classi
- Le classi possono avere relazioni fra loro, rappresentate con le *associazioni*
- Per default, un'associazione è bidirezionale, anche se può essere resa unidirezionale

Ing. SW: il ruolo di UML - 92

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Rappresentazione di una classe: 1) come solo entità

---

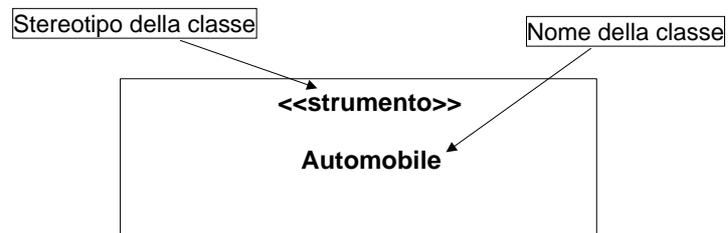


Ing. SW: il ruolo di UML - 93

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Rappresentazione di una classe: 1a) come entità + stereotipo

---

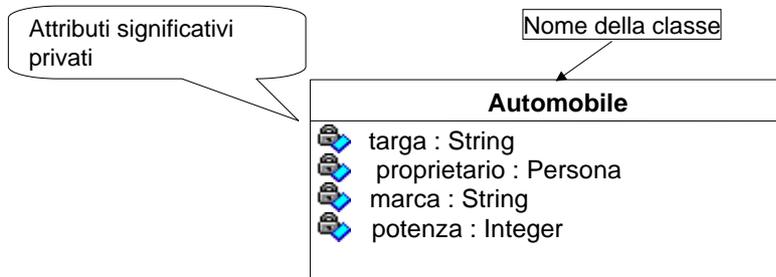


Ing. SW: il ruolo di UML - 94

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Rappresentazione di una classe: 2) entità con attributi

---

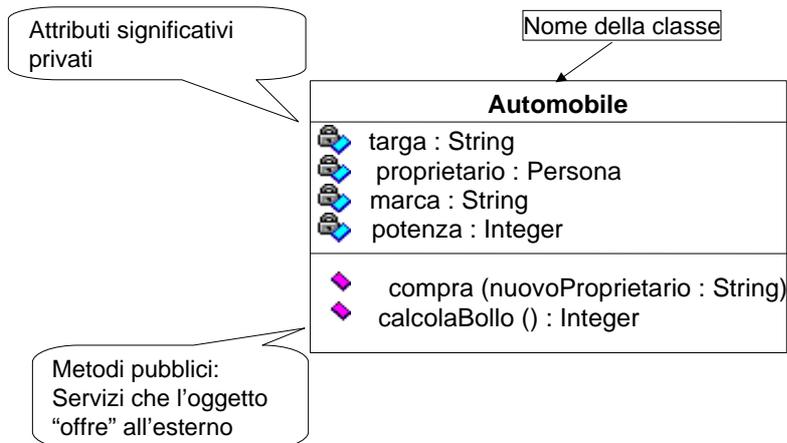


Ing. SW: il ruolo di UML - 95

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Rappresentazione di una classe: 3) entità + attributi + metodi

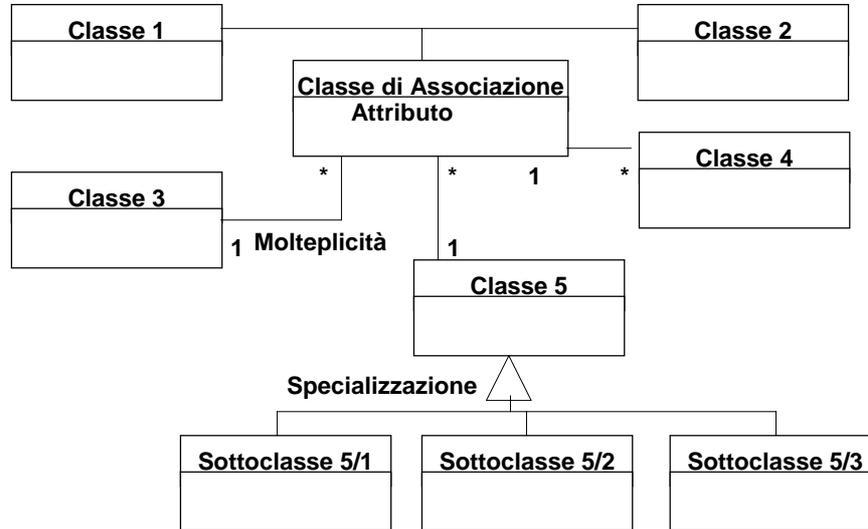
---



Ing. SW: il ruolo di UML - 96

Giulio Destri - © AreaSP for Univ. Parma, 2006

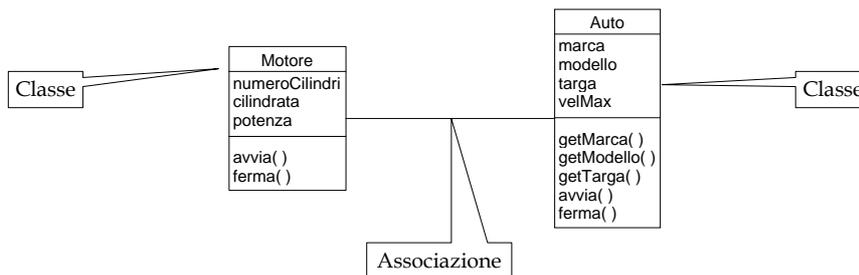
## Diagramma delle classi



Ing. SW: il ruolo di UML - 97

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un esempio di associazione



Fra le classi Auto e Motore c'è un'associazione: questo significa che fra ogni istanza della classe auto e ogni istanza della classe motore c'è un *link*

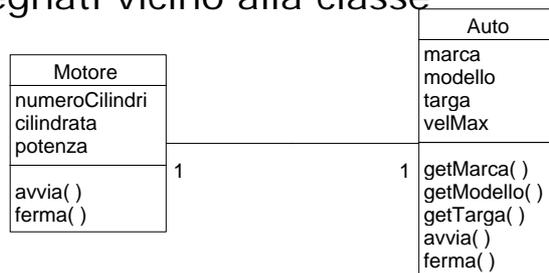
Oggetto : classe = link : associazione

Ing. SW: il ruolo di UML - 98

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Cardinalità

- La *cardinalità* di un'associazione esprime il numero di oggetti di una certa classe che prendono parte all'associazione
- Si esprime con un numero o un range disegnati vicino alla classe

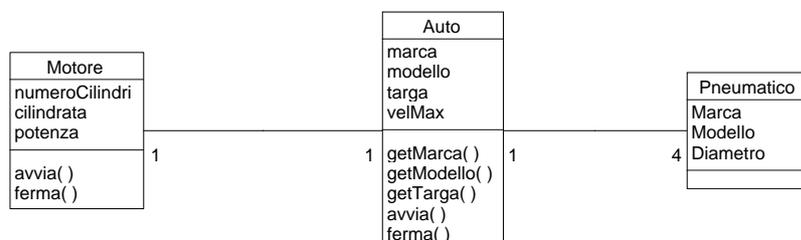


Ing. SW: il ruolo di UML - 99

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Cardinalità - 2

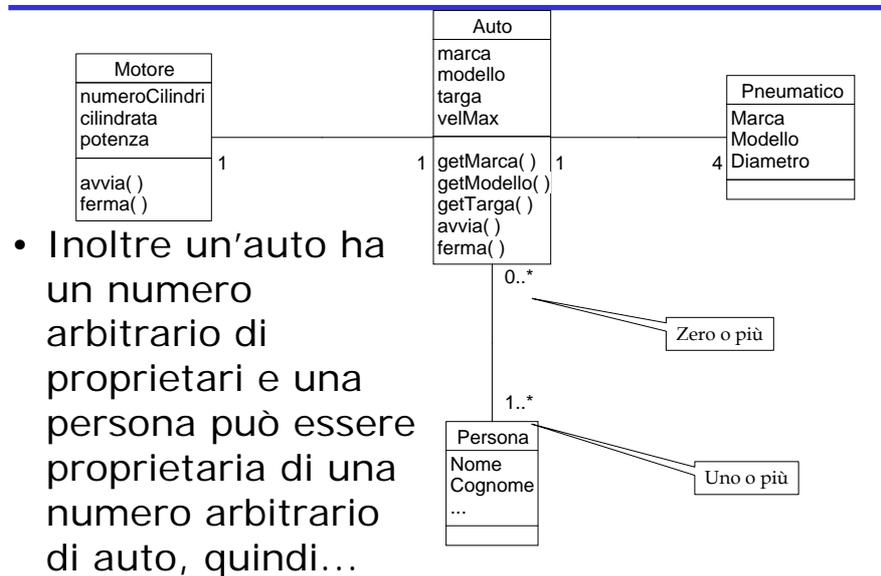
- Un auto ha 4 pneumatici, per cui il diagramma diventa



Ing. SW: il ruolo di UML - 100

Giulio Destri - © AreaSP for Univ. Parma, 2006

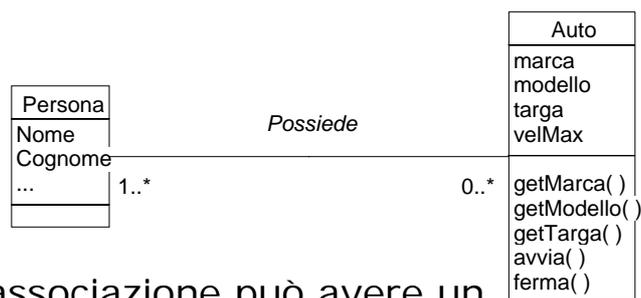
## Cardinalità - 3



Ing. SW: il ruolo di UML - 101

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Nomi per le associazioni



- Un'associazione può avere un nome
- Il concetto di "possiede" è però unidirezionale

Ing. SW: il ruolo di UML - 102

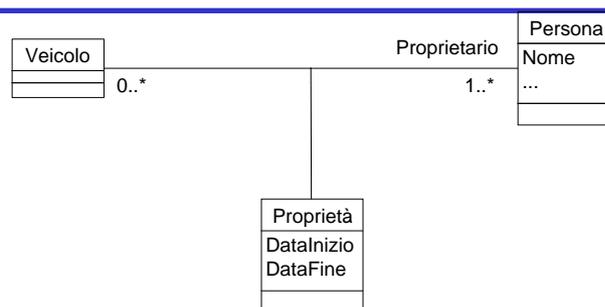
Giulio Destri - © AreaSP for Univ. Parma, 2006

## Ruoli per le classi



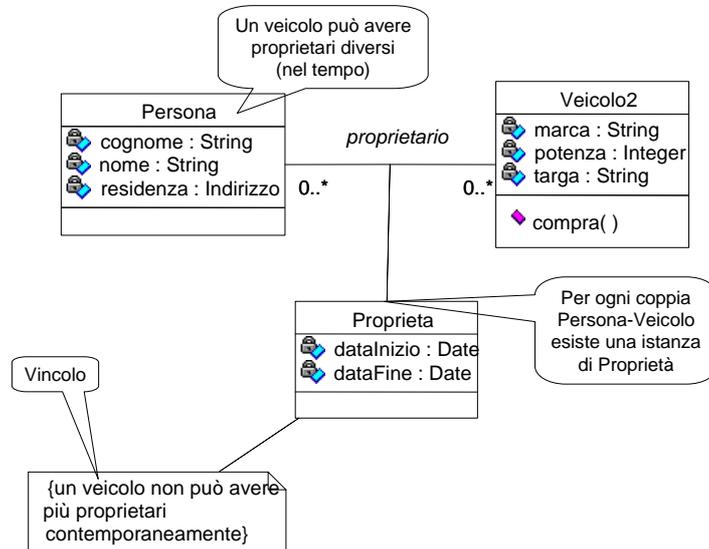
In genere si preferisce usare  
Ruoli associati a una delle classi

## Classi di associazione



- Servono ad esprimere attributi e/o metodi propri dell'associazione
- I concetti espressi non appartengono agli oggetti associati ma all'associazione in quanto tale

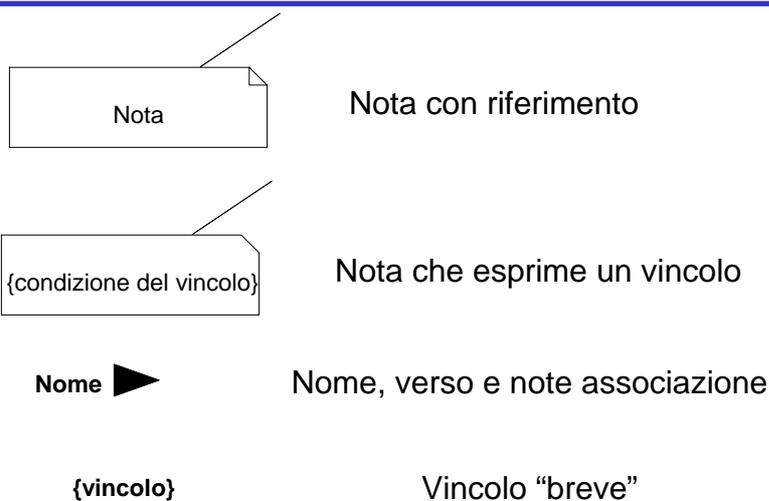
## Classi di associazione e vincoli



Ing. SW: il ruolo di UML - 105

Giulio Destri - © AreaSP for Univ. Parma, 2006

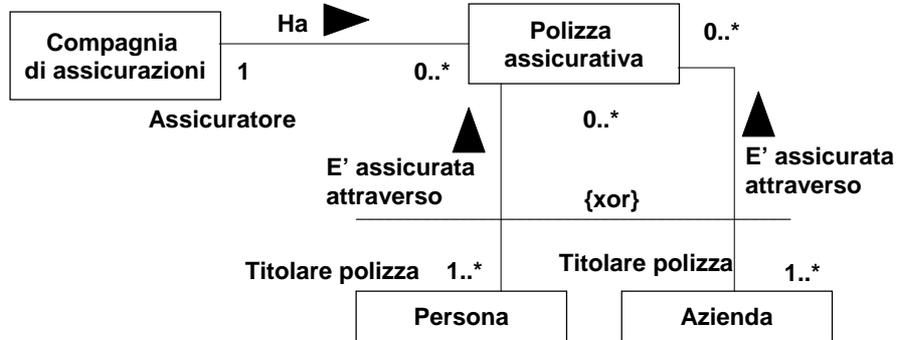
## Vincoli e note



Ing. SW: il ruolo di UML - 106

Giulio Destri - © AreaSP for Univ. Parma, 2006

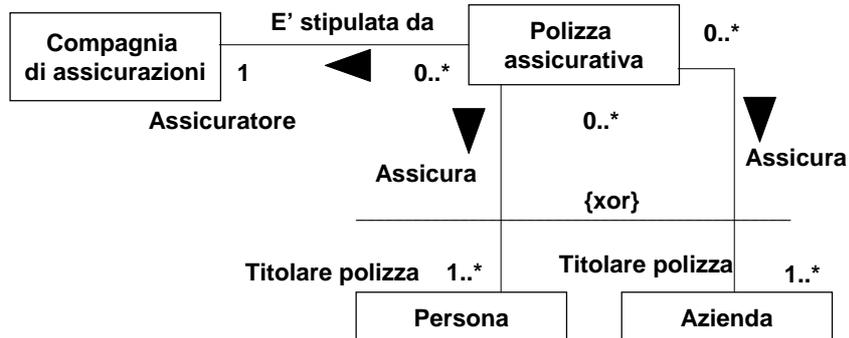
## Dettagli delle associazioni: le relazioni



Ing. SW: il ruolo di UML - 107

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Dettagli delle associazioni: le relazioni



Ing. SW: il ruolo di UML - 108

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Class diagram e diagrammi E-R

---

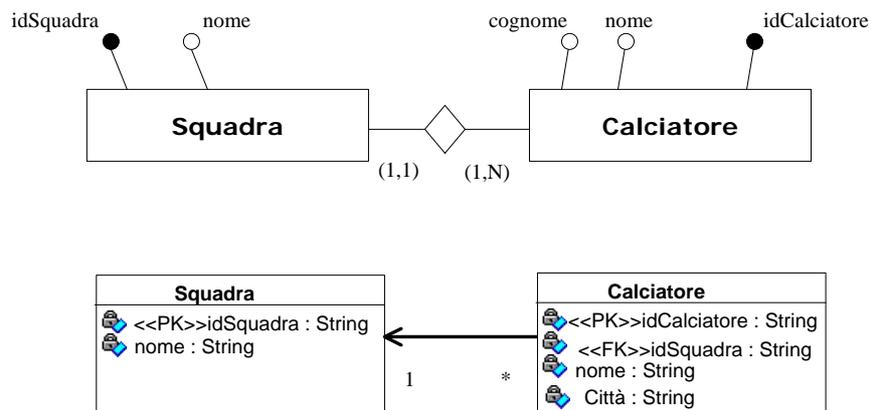
- I diagrammi di classe rappresentano una evoluzione dei diagrammi E-R
- Vengono inseriti concetti non direttamente rappresentabili negli E-R, quali:
  - Inclusione
  - Ereditarietà
  - Operazioni (metodi) associate alle classi

Ing. SW: il ruolo di UML - 109

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Class diagram e diagrammi E-R: esempio

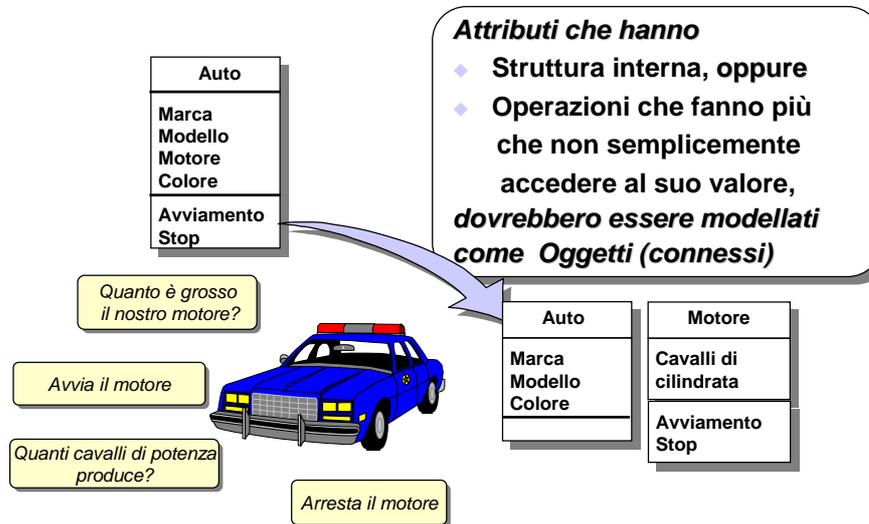
---



Ing. SW: il ruolo di UML - 110

Giulio Destri - © AreaSP for Univ. Parma, 2006

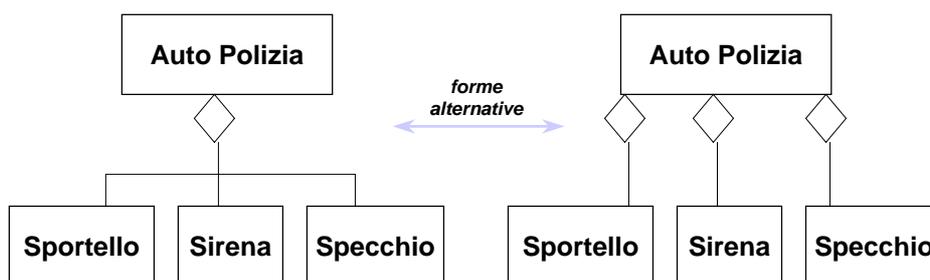
## Cosa accade quando gli Attributi diventano complessi?



Ing. SW: il ruolo di UML - 111

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Aggregazione di classi



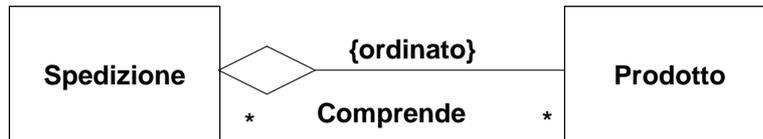
- Il diamante o rombo che tocca il simbolo di una classe indica che essa contiene gli altri elementi (è un aggregato)
- La linea di relazione conduce dal diamante alle classi componenti
- Forme gerarchiche e multi-diamante hanno lo stesso significato

Ing. SW: il ruolo di UML - 112

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Diagramma delle classi: l'aggregazione

---



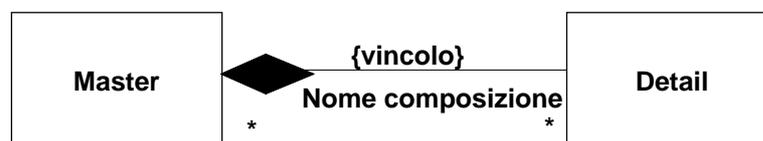
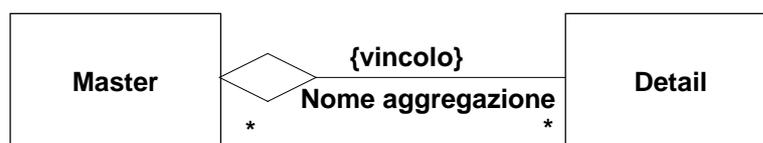
Una spedizione comprende un certo numero di prodotti (quelli ordinati)  
Un prodotto può essere compreso in più di una spedizione

Ing. SW: il ruolo di UML - 113

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Aggregazione e composizione

---



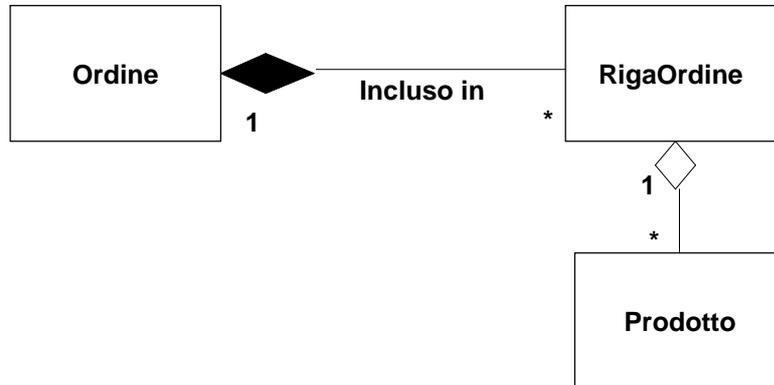
- L'eliminazione di una composizione (diamante nero) elimina anche tutti i suoi elementi componenti
- L'eliminazione di una aggregazione (diamante bianco) invece no (i componenti hanno anche una natura Indipendente)

Ing. SW: il ruolo di UML - 114

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Aggregazione e composizione: esempio

---



**L'eliminazione di un ordine elimina anche le righe ordine ma non elimina i prodotti che esse comprendono**

## Le Generalizzazioni

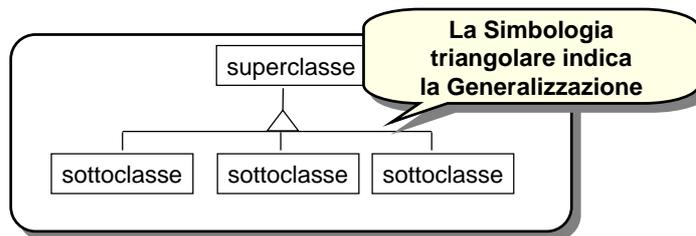
---

- Le Generalizzazioni sfruttano le Analogie delle Classi, proprio come le Classi sfruttano le Analogie degli Oggetti
- Le Generalizzazioni sfruttano gli elementi in comune delle Classi
- Le Specializzazioni definiscono le Differenze
- Le Gerarchie di Generalizzazione definiscono una "specie" di relazione

## La Generalizzazione ha una Simbologia Speciale

---

- Rapporti delle Generalizzazioni
  - La classe da specificare è una superclasse
  - La classe specificata è una sottoclasse
  - I genitori si riferiscono all'insieme di tutte le superclassi
  - I figli si riferiscono all'insieme di tutte le sottoclassi

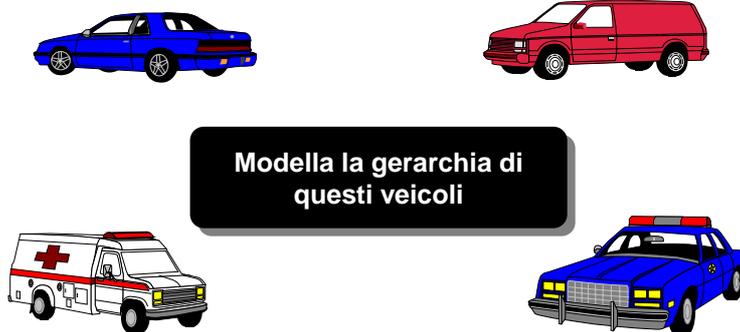


Ing. SW: il ruolo di UML - 117

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Cosa è analogo in questi Oggetti?

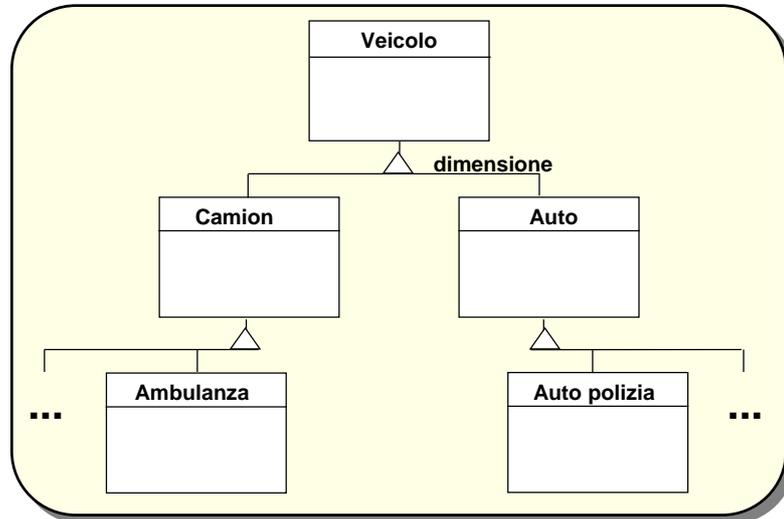
---



Ing. SW: il ruolo di UML - 118

Giulio Destri - © AreaSP for Univ. Parma, 2006

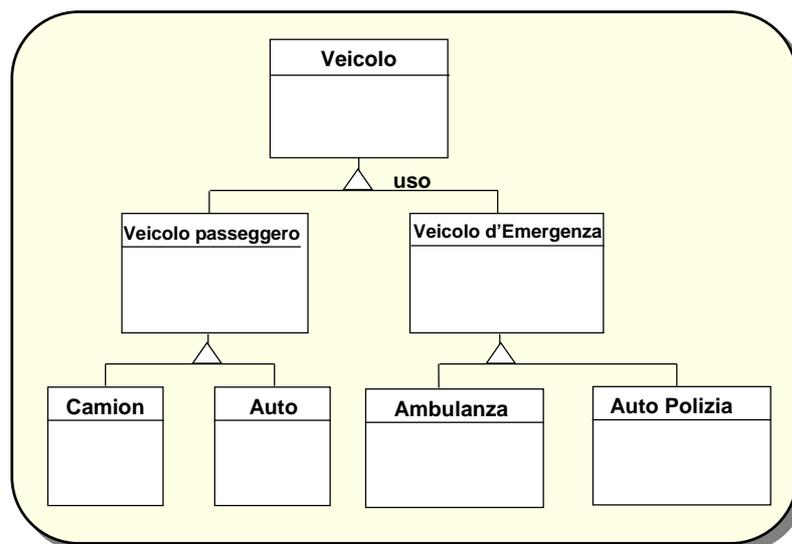
## Una possibile gerarchia



Ing. SW: il ruolo di UML - 119

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un'altra gerarchia



Ing. SW: il ruolo di UML - 120

Giulio Destri - © AreaSP for Univ. Parma, 2006

## II Discriminante

---

- Fornisce le Basi per la Specializzazione specificando quale proprietà della "superclasse" è da astrarre
- Valori opzionali, enumerati
- Attributo speciale che in altro modo non appare sul diagramma

## II Discriminante

---

- Solo una proprietà alla volta dovrebbe essere distinta
- Ogni sottoclasse ha un valore unico per il discriminante
  - Si distingue dalle altre sottoclassi

## Le Sottoclassi ereditano Proprietà dai Genitori

---

- Le Sottoclassi ereditano gli Attributi, le operazioni e le Associazioni delle loro superclassi
  - Entrambe le sottoclassi ereditano il Tipo Sirena e il Suono Sirena
- L'esempio di una sottoclasse è un esempio di tutte le classi genitrici
- Specializzazioni delle sottoclassi tramite l'aggiunta di proprietà uniche
  - L'ambulanza aggiunge il carico della vittima, il veicolo della Polizia aggiunge il carico del criminale

Ing. SW: il ruolo di UML - 123

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le Sottoclassi ereditano Proprietà dai Genitori - 2

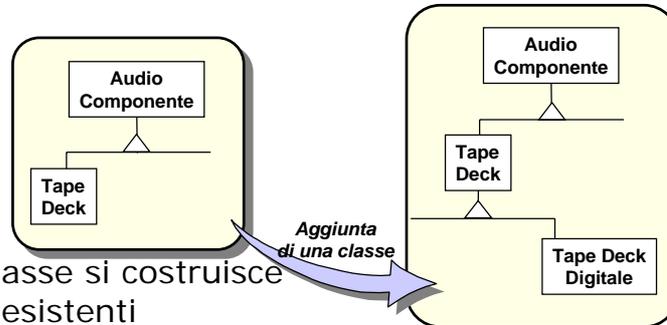
---



Ing. SW: il ruolo di UML - 124

Giulio Destri - © AreaSP for Univ. Parma, 2006

## L'ereditarietà ci consente di costruire da Componenti Simili

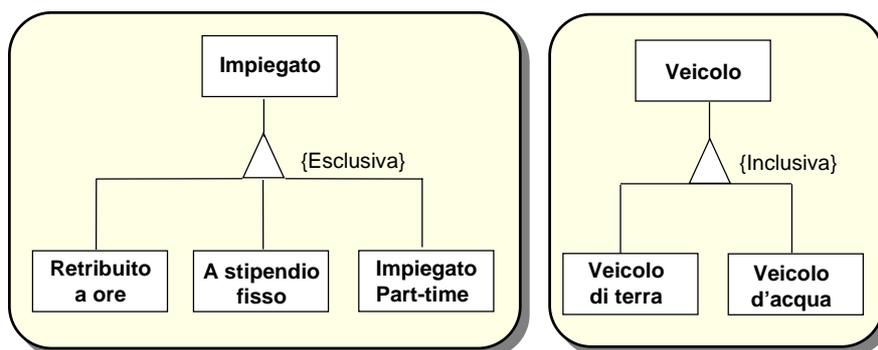


- La nuova classe si costruisce sulle classi esistenti ereditando le proprietà dei genitori nella gerarchia di generalizzazione
- La nuova classe ha bisogno solo d'implementare le estensioni e le differenze

Ing. SW: il ruolo di UML - 125

Giulio Destri - © AreaSP for Univ. Parma, 2006

## La suddivisione in Sottoclassi può essere Esclusiva o Inclusiva

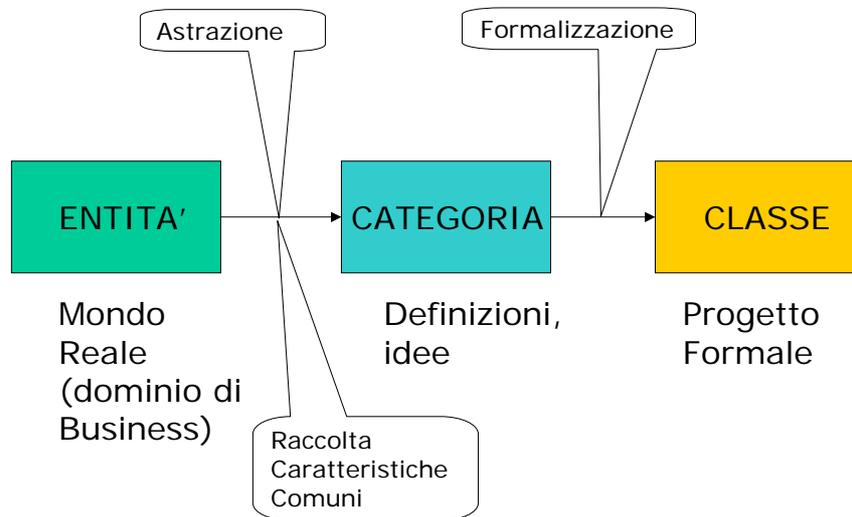


- L'impiegato è retribuito a ore, a stipendio fisso o part time
- Un veicolo è un veicolo di terra, un veicolo d'acqua o entrambi.

Ing. SW: il ruolo di UML - 126

Giulio Destri - © AreaSP for Univ. Parma, 2006

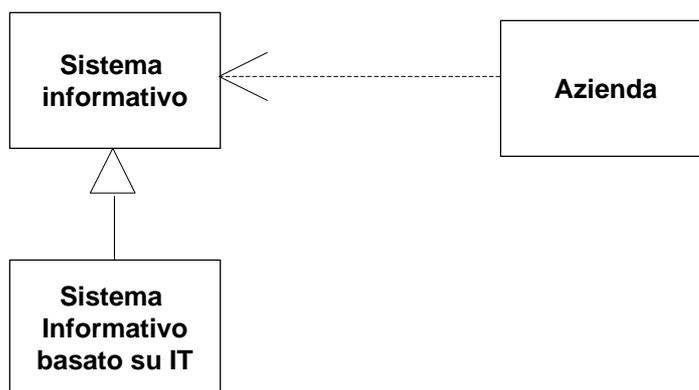
## Il passaggio da entità a classi



Ing. SW: il ruolo di UML - 127

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Diagramma delle classi: dipendenza



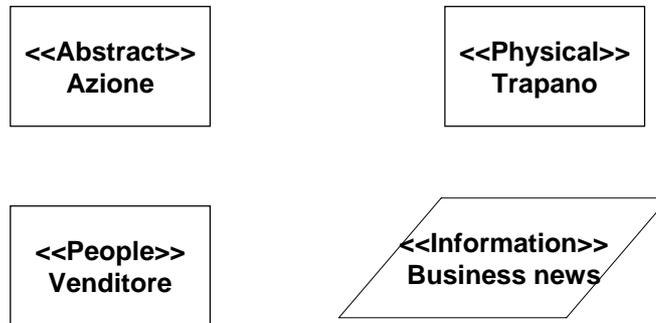
**L'Azienda dipende dal sistema informativo, che viene Realizzato attraverso un sistema informativo basato sull'information technology**

Ing. SW: il ruolo di UML - 128

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Classi e risorse

---



Gli stereotipi definiscono concetti che personalizzano le entità modellizzate con le classi

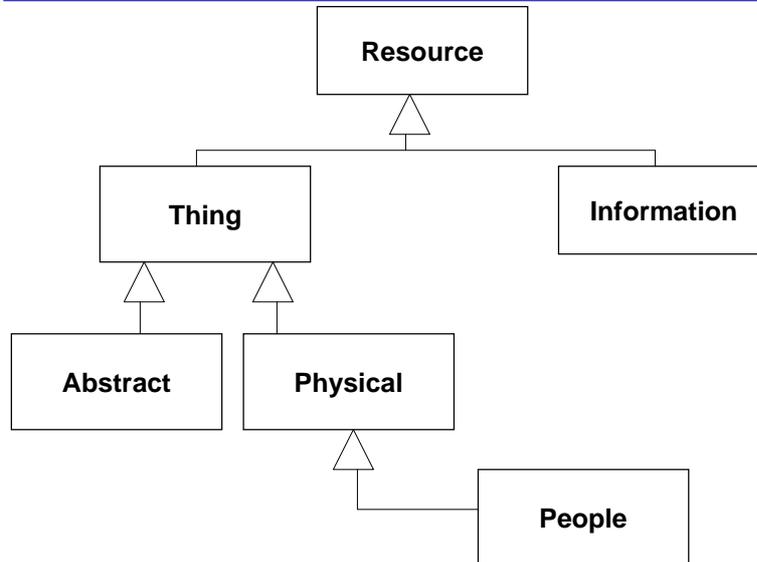
## Classi e risorse: metamodelli

---

- Attraverso i class diagram è possibile esprimere anche concetti costitutivi di altre classi e/o categorie
- Si possono costruire quindi dei metamodelli

## Metamodelli: esempio

---



Ing. SW: il ruolo di UML - 131

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Class diagram ed object diagram

---

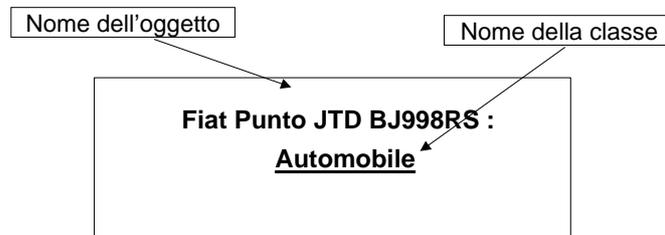
- Gli object diagram sono simili ai class diagram, ma i loro componenti sono gli oggetti, ossia istanze ben definite delle classi
- In pratica quindi un object diagram rappresenta un insieme di legami logici caratteristici di di entità concrete e non astratte

Ing. SW: il ruolo di UML - 132

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Object diagram: sintassi

---

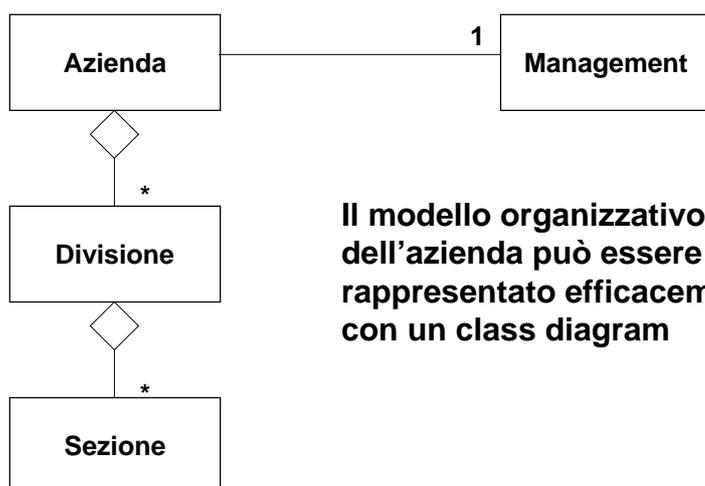


Ing. SW: il ruolo di UML - 133

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Classi ed oggetti: organizzazione

---

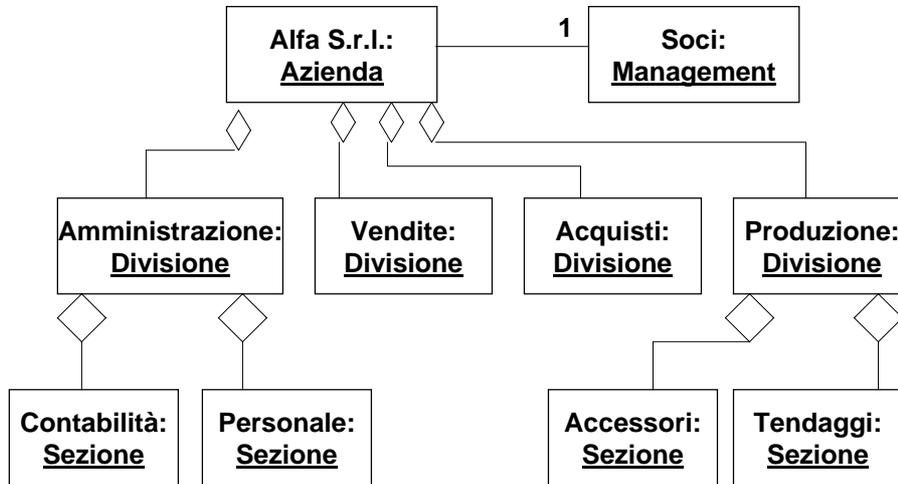


**Il modello organizzativo dell'azienda può essere rappresentato efficacemente con un class diagram**

Ing. SW: il ruolo di UML - 134

Giulio Destri - © AreaSP for Univ. Parma, 2006

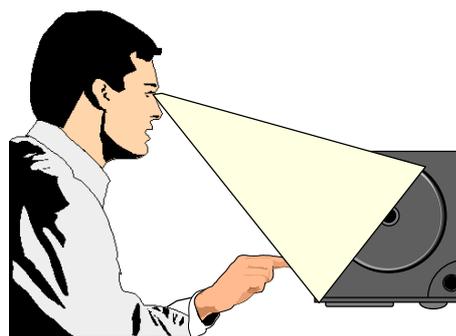
## Organizzazione effettiva: object diagram



Ing. SW: il ruolo di UML - 135

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio generale: messaggi e metodi

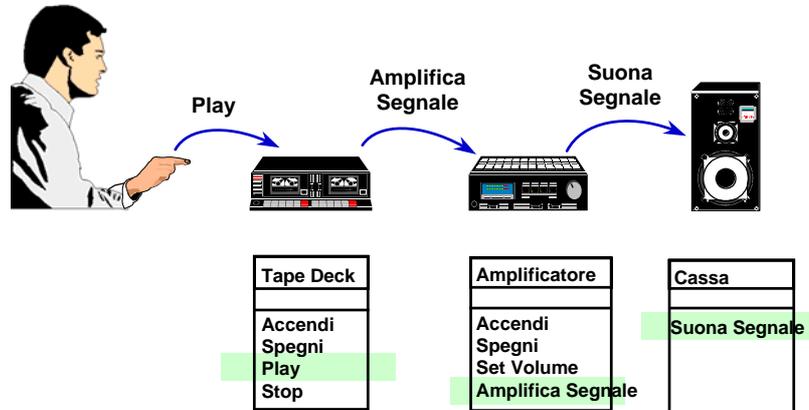


Accendi, Spegni,  
Play, Pause, Stop

Ing. SW: il ruolo di UML - 136

Giulio Destri - © AreaSP for Univ. Parma, 2006

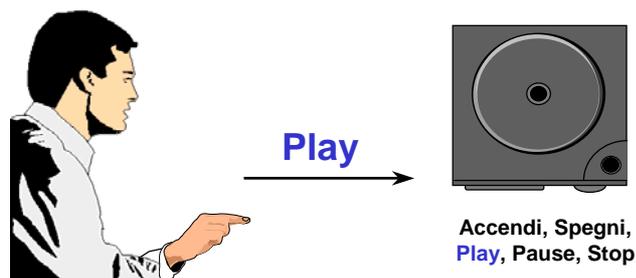
## Oggetti comunicano inviando Messaggi



Ing. SW: il ruolo di UML - 137

Giulio Destri - © AreaSP for Univ. Parma, 2006

## I Messaggi sono richieste di servizio



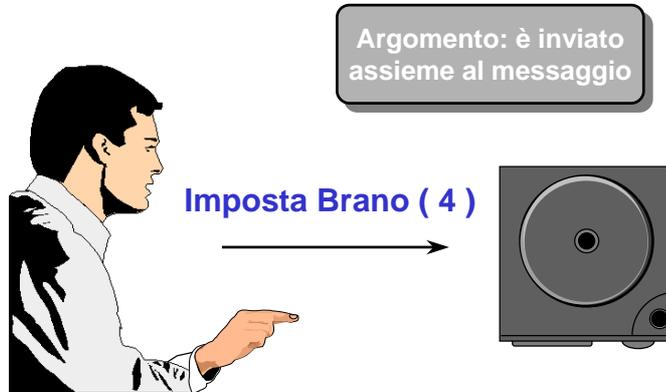
- Gli oggetti comunicano inviando Messaggi
- Le "Message signatures" identificano l'operazione richiesta
- Oggetto ricevente risponde eseguendo l'operazione corrispondente

Ing. SW: il ruolo di UML - 138

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Messaggi possono avere Argomenti e Informazioni di Ritorno

---

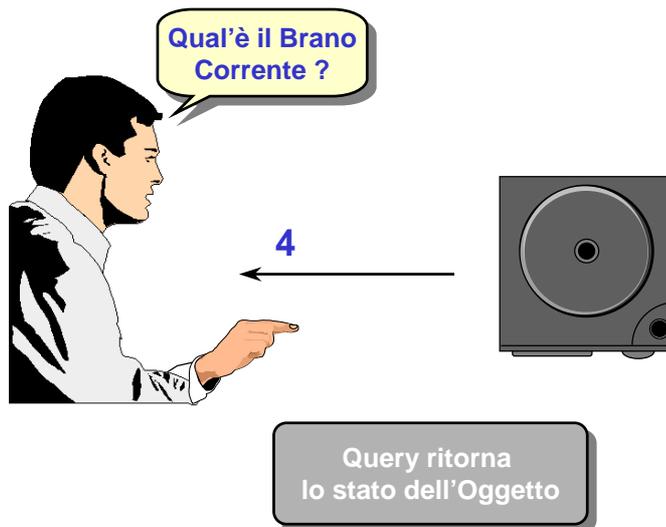


Ing. SW: il ruolo di UML - 139

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Messaggi possono avere Argomenti e Informazioni di Ritorno

---

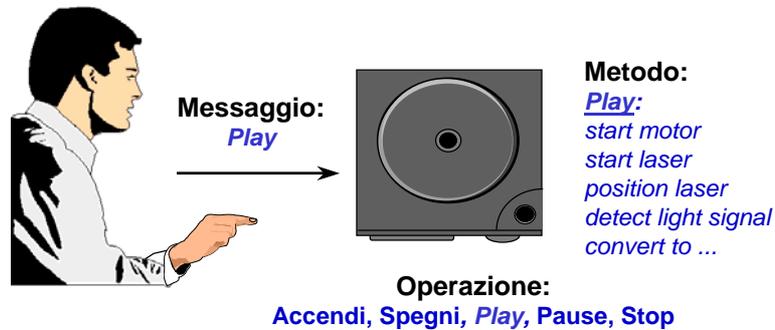


Ing. SW: il ruolo di UML - 140

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Metodo: definisce *come* un oggetto Esegue un operazione

---



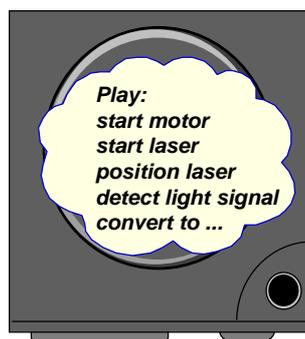
Messaggio	segnale tra oggetti
Operazione	servizio supportato da un oggetto
Metodo	implementazione di un'operazione

Ing. SW: il ruolo di UML - 141

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Metodi sono nascosti dentro gli oggetti

---



- Separazione di Metodi dalle Operazioni
  - Migliora la portabilità
  - Riduce l'effetto ricaduta delle modifiche
- Metodi sono nascosti agli altri oggetti
- Operazioni sono interfacce visibili verso i Metodi

Ing. SW: il ruolo di UML - 142

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Molte classi possono fornire lo stesso servizio in modi diversi

---



**Polimorfismo**: abilità di due o più classi di oggetti di rispondere allo stesso messaggio, ognuno "a suo modo"



- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Statechart Diagram

## Diagrammi di sequenza

---

- Descrivono le interazioni fra gli oggetti organizzate in sequenza temporale
- Uno use case contiene al suo interno vari diagrammi di sequenza
- Salvo casi banali, non si rappresentano all'inizio tutte le possibili sequenze, ma solo le principali

## Diagrammi di sequenza: gli elementi

---

- Elementi costitutivi di un diagramma di sequenza:
  - gli oggetti
  - i messaggi attraverso cui essi interagiscono.
- Lo scambio di messaggi è rappresentato da frecce con un nome.

## Diagrammi di sequenza: i messaggi

---

In base al livello di dettaglio

- I messaggi possono esplicitamente far riferimento ai metodi (operazioni) effettivamente coinvolti, ossia richiamati;
- Possono essere specificati anche i parametri e/o i risultati;
- Oppure viene descritta un'azione generica.

## Diagrammi di sequenza: il controllo

---

- Ripetizioni cicliche (for, while)
- Condizioni (if, case)

## Diagrammi di sequenza: messaggi e azioni

---

- **Call**: invoca un metodo di un oggetto; un oggetto può inviare un messaggio a se stesso, invocando un proprio metodo
- **Return**: restituisce un valore al chiamante
- **Send**: invia un signal ad un oggetto
- **Create**: crea un oggetto
- **Destroy**: distrugge un oggetto; un oggetto può distruggere se stesso

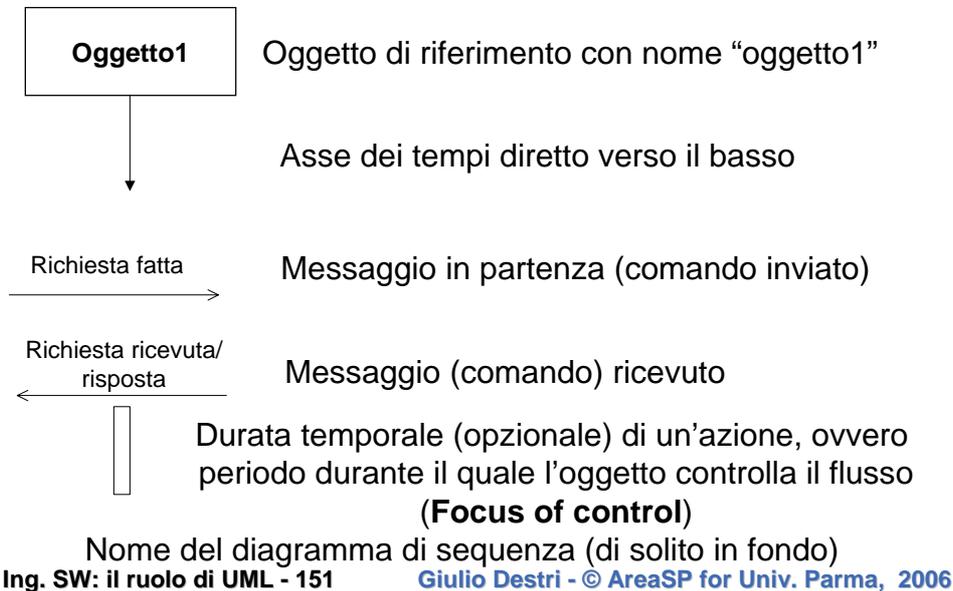
## Diagrammi di sequenza: sequenze di messaggi

---

- Uno scambio di messaggi può dare origine ad una sequenza
- La sequenza deve avere un punto d'inizio ("evento scatenante")
- Può essere utile anteporre numeri ai nomi dei messaggi

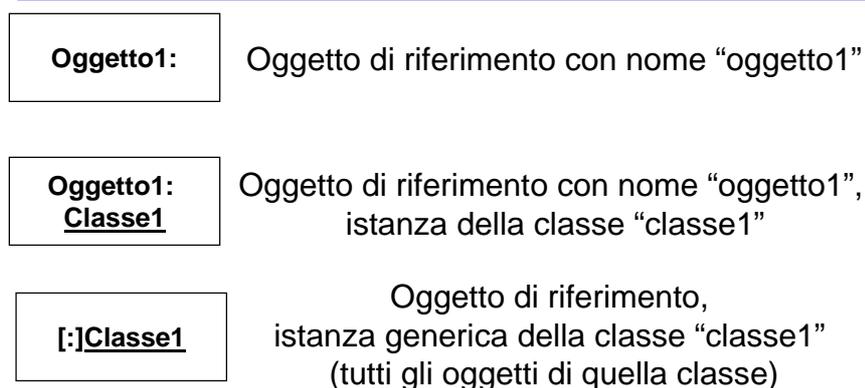
## Sequence Diagram: simboli base

---



## Sequence Diagram: tipi di entità

---



## Sequence Diagram: tipi di messaggi

---

—————> **Semplice**: il controllo è passato dal chiamante al ricevente

—————> **Sincrono**: il controllo è passato dal chiamante al ricevente ed il primo attende che il secondo gli restituisca il controllo

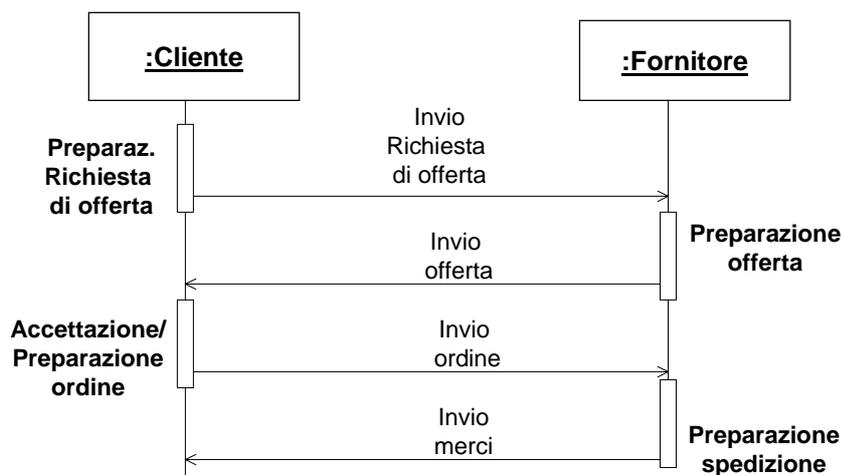
—————> **Asincrono**: il chiamante trasmette un segnale al ricevente ma prosegue poi nelle proprie azioni senza attendere il secondo che può o meno ritornare informazioni

Ing. SW: il ruolo di UML - 153

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio di diagramma di sequenza: offerta

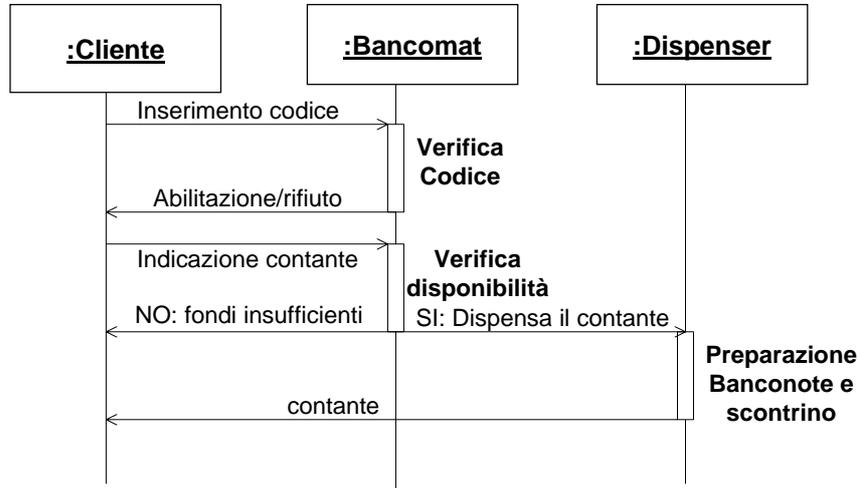
---



Ing. SW: il ruolo di UML - 154

Giulio Destri - © AreaSP for Univ. Parma, 2006

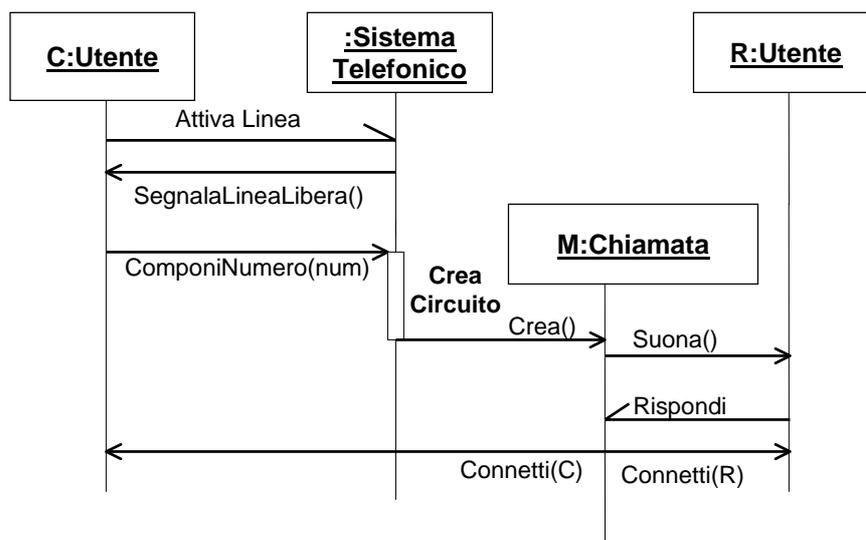
## Esempio di diagramma di sequenza: bancomat



Ing. SW: il ruolo di UML - 155

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio di diagramma di sequenza: telefono



Ing. SW: il ruolo di UML - 156

Giulio Destri - © AreaSP for Univ. Parma, 2006

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- **Collaboration Diagram**
- Activity Diagram
- Statechart Diagram

## Diagrammi di collaborazione

---

- Sono semanticamente equivalenti ai diagrammi di sequenza: entrambi sono visualizzazioni di scenari
- I diagrammi di collaborazione enfatizzano le relazioni fra oggetti (ovvero l'organizzazione strutturale), i diagrammi di sequenza enfatizzano la sequenza temporale delle comunicazioni

## Diagrammi di collaborazione - 2

---

- La sequenza dei messaggi è meno evidente che nel diagramma di sequenza, mentre sono più evidenti i legami tra gli oggetti
- I messaggi hanno sempre espresso l'ordine di sequenza

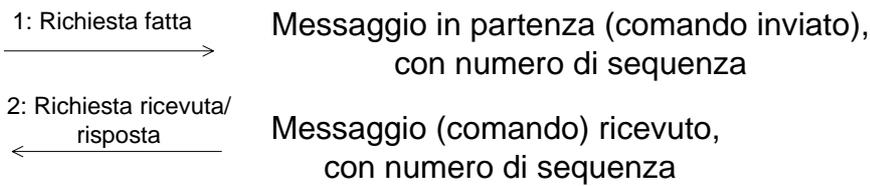
## Diagrammi di collaborazione - 3

---

- I diagrammi di collaborazione vengono usati prevalentemente in fase di progetto, quelli di sequenza in fase di analisi, perché sono più comprensibili da parte del committente (cliente, esperto del dominio)
- I due diagrammi sono isomorfi, è possibile cioè trasformare uno nell'altro

## Collaboration Diagram: simboli base

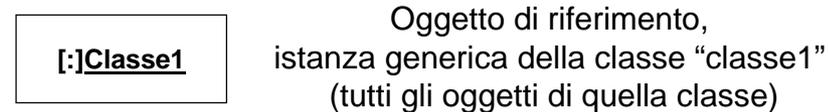
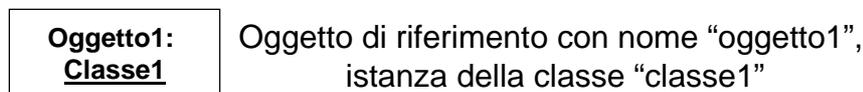
---



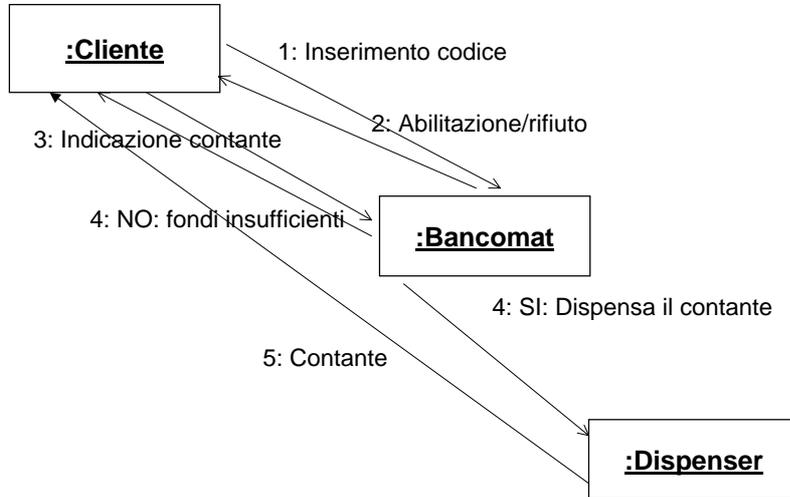
Nome del diagramma di collaborazione (di solito in fondo)

## Collaboration Diagram: tipi di entità

---



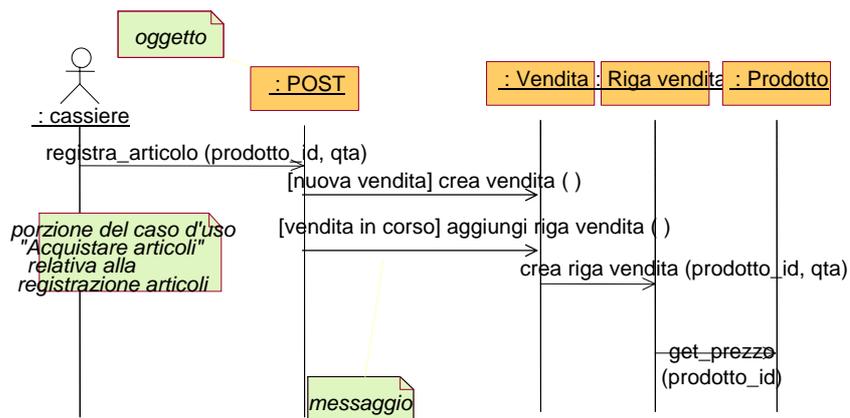
## Esempio di diagramma di collaborazione: bancomat



Ing. SW: il ruolo di UML - 163

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio completo: sequenza

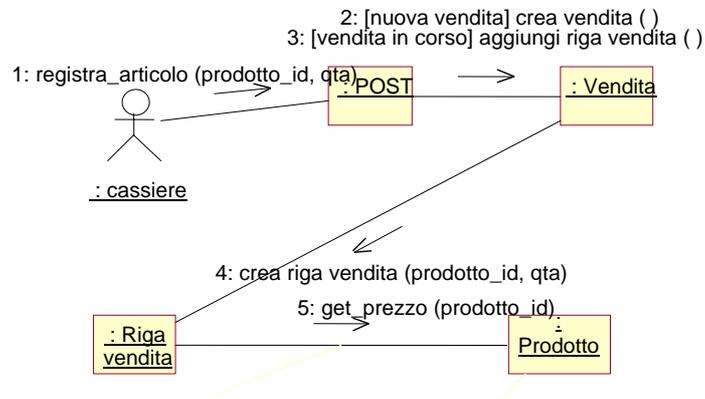


Ing. SW: il ruolo di UML - 164

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio completo: collaborazione

---



Ing. SW: il ruolo di UML - 165

Giulio Destri - © AreaSP for Univ. Parma, 2006



- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- **Activity Diagram**
- Statechart Diagram

Ing. SW: il ruolo di UML - 166

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram (diagrammi d'attività)

---

- Rappresentano una procedura o un workflow
- Mostrando l'evoluzione di un flusso di attività
- Ogni attività è definita come un'evoluzione continua, non necessariamente atomica, di uno stato
- Sono una evoluzione dei flow-chart

## Activity Diagram: elementi base

---

- Activity (Attività)  
Esecuzione non atomica entro un sistema dotato di stati.  
Può essere scomposta in azioni.
- Action (Azione)  
Operazione atomica eseguibile che produce come risultato un cambiamento nello stato di un sistema o il ritorno di un valore.

## Activity Diagram: concetti base

---

- **Action State (Stato di azione)**  
Uno stato che rappresenta l'esecuzione di un'azione (atomica), tipicamente l'invocazione di una operazione.
- **Activity State (Stato di attività)**  
Stato composto, in cui il flusso di controllo è formato di altri stati di attività e stati di azione. Non è atomico, il che significa anche che può essere interrotto. Può anche essere ulteriormente scomposto in altri diagrammi di attività

Ing. SW: il ruolo di UML - 169

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: elementi di contorno

---

- **Transition (Transizione)**  
Rappresenta il flusso di controllo fra due attività, che mostra il percorso da un action o activity state al successivo action o activity state.
- **Object Flow**  
Rappresenta un oggetto (un'entità) coinvolta nel flusso di controllo associato con un activity diagram.

Ing. SW: il ruolo di UML - 170

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: elementi di contorno

---

- Object State  
Una condizione o situazione operativa nella vita di un oggetto (un'entità) durante la quale l'oggetto soddisfa certe condizioni, compie certe attività o attende certi eventi.
- Swimlane  
Una suddivisione per l'organizzazione di responsabilità per le attività. Non ha un significato fisso, ma spesso corrisponde alla unità organizzativa entro un business model (es. ufficio acquisti, vendite...).

Ing. SW: il ruolo di UML - 171

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: tipologie

---

- Activity state (Diagram)  
Le singole attività hanno una durata e possono essere ulteriormente scomposte, dando origine ad altri diagrammi
- Action state (Diagram)  
Le singole attività sono atomiche e non possono essere ulteriormente scomposte

Ing. SW: il ruolo di UML - 172

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: significato

---

- Enfasi posta sulle attività e non su chi le compie
- Enfasi sulla sequenza di azioni di una particolare procedura
- Vengono evidenziati vincoli di precedenza o di concorrenza

Ing. SW: il ruolo di UML - 173

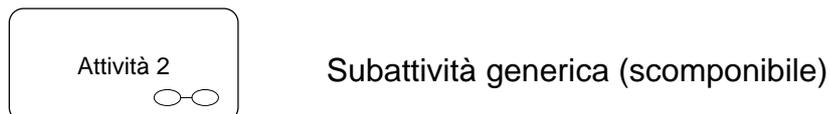
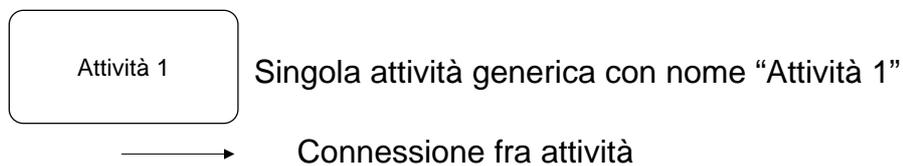
Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: simboli base

---

 Inizio di un diagramma di attività

 Termine di un diagramma di attività



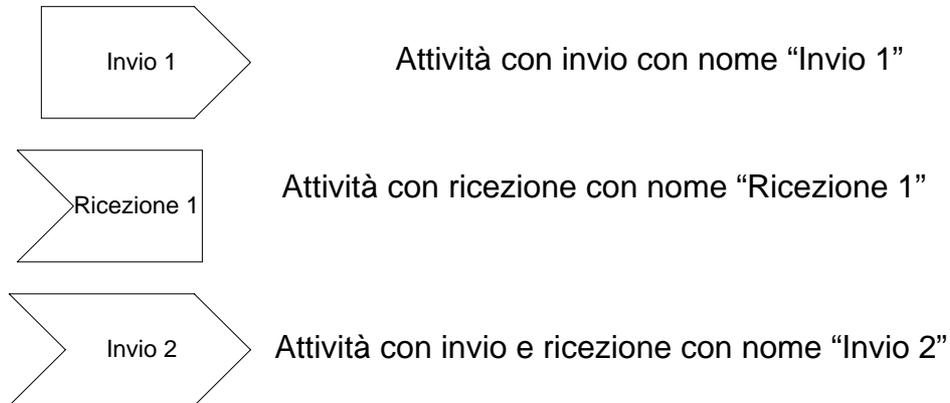
Non vi è differenza fra simboli di attività e simboli di azione  
Nome del diagramma di attività (in fondo)

Ing. SW: il ruolo di UML - 174

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram: simboli di invio/ricezione

---



Ing. SW: il ruolo di UML - 175

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: sequenza semplice

---



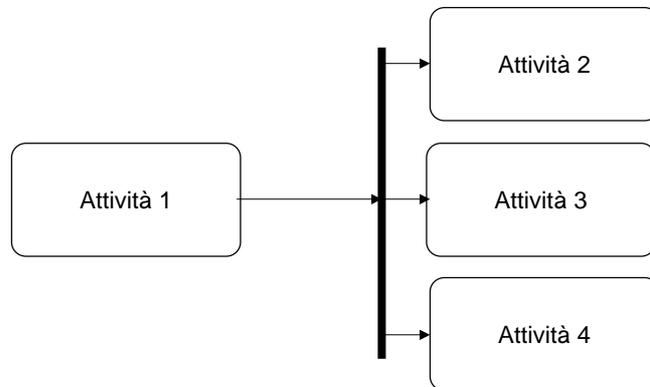
Un'attività (Attività 2) viene eseguita dopo la fine della  
Precedente (Attività 1)  
**(Single Thread)**

Ing. SW: il ruolo di UML - 176

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: AND-split

---



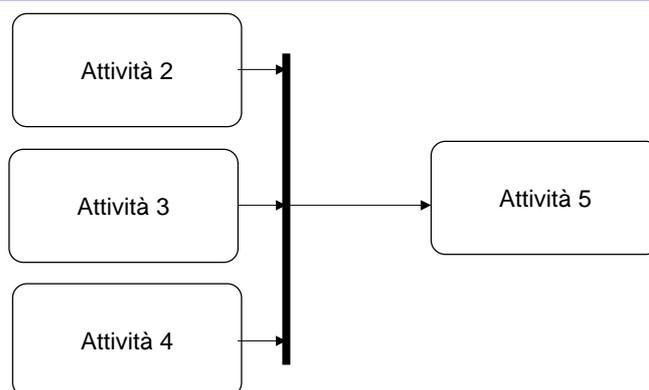
Un singolo flusso di attività si divide in più flussi, consentendo l'esecuzione simultanea di più attività  
**(Multiple Thread)**

Ing. SW: il ruolo di UML - 177

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: AND-join

---



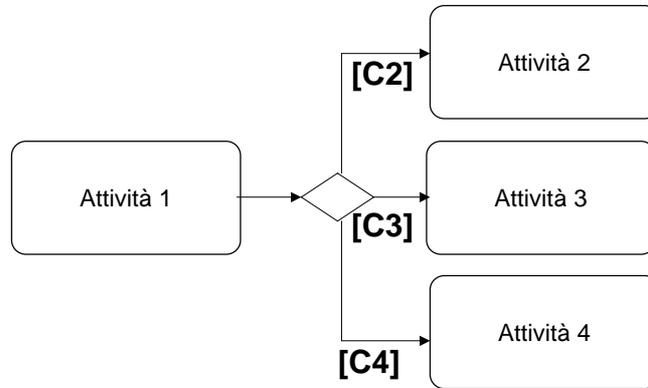
Due o più flussi di attività convergono in uno solo  
E' un punto di sincronizzazione per il workflow: non si va avanti finché non sono terminate tutte le attività precedenti

Ing. SW: il ruolo di UML - 178

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: OR-split

---



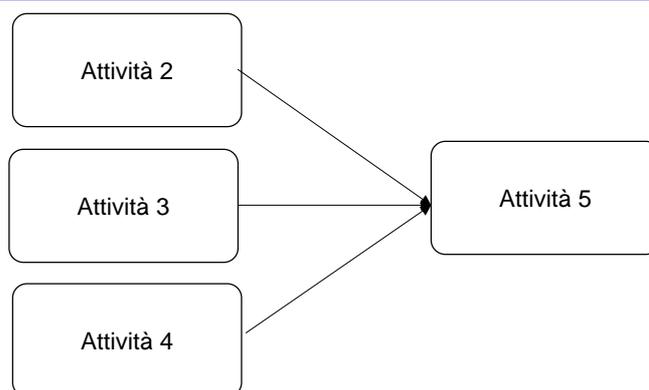
Un singolo flusso di attività prosegue per uno dei cammini in base al verificarsi delle condizioni di transizione, indicate fra parentesi quadre

Ing. SW: il ruolo di UML - 179

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: OR-join

---



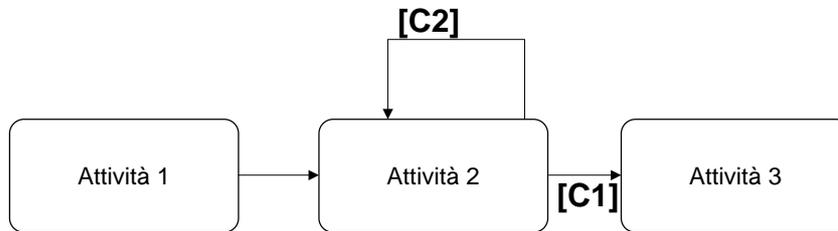
Un punto dove due o più flussi di attività ri-convergono in uno solo ovvero hanno tutti Attività 5 come elemento successivo

Ing. SW: il ruolo di UML - 180

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di AD: iterazione

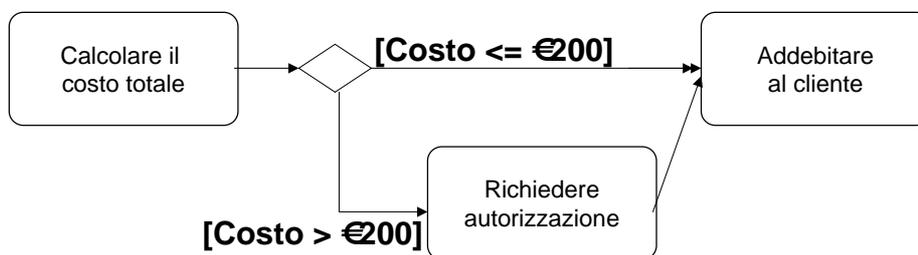
---



Un'attività (Attività 2) viene ripetuta più volte, in base al verificarsi o meno di opportune condizioni di controllo

## Activity Diagram: esempi vari

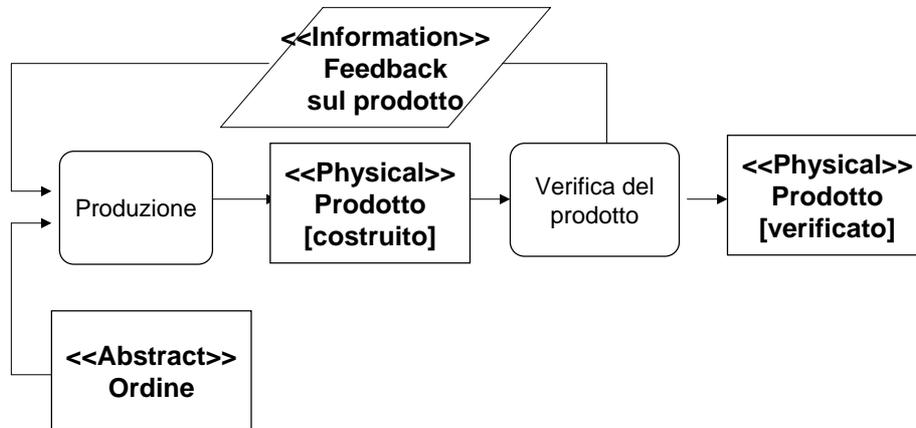
---



Se il costo totale è maggiore di 200€, bisogna chiedere l'autorizzazione prima di addebitarlo al cliente.

## Activity Diagram: flusso di oggetti

---

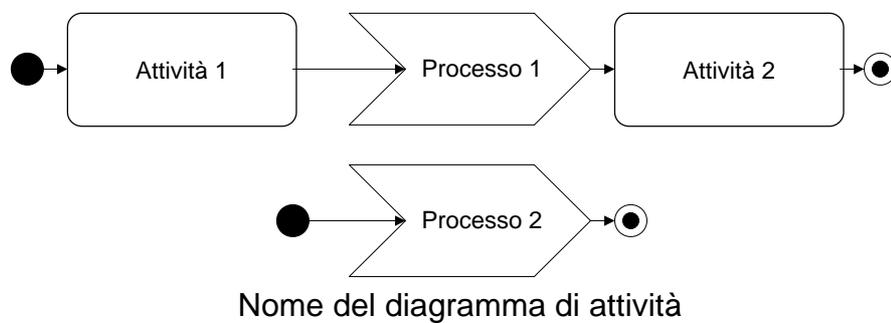


Ing. SW: il ruolo di UML - 183

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram con processi

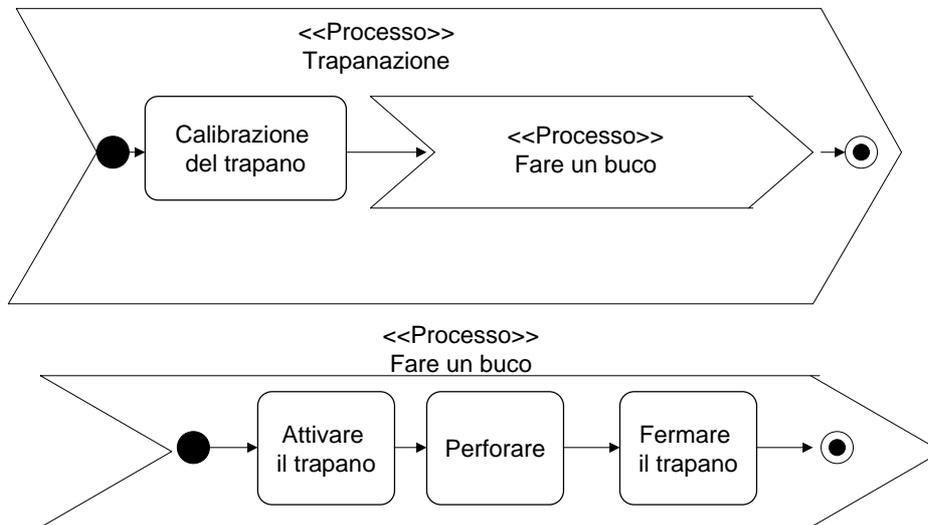
---



Ing. SW: il ruolo di UML - 184

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Activity Diagram con processi: esempio



Ing. SW: il ruolo di UML - 185

Giulio Destri - © AreaSP for Univ. Parma, 2006



- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Statechart Diagram

Ing. SW: il ruolo di UML - 186

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Statechart Diagram: diagrammi di stato

---

- Possono essere usati per descrivere il comportamento nel tempo di un particolare elemento come
- un oggetto (ovvero una singola entità)
- un intero sottosistema
- ovvero l'evoluzione di una interazione.

## Statechart Diagram: diagrammi di stato

---

In pratica essi descrivono

- sequenze di **stati** ed **azioni** attraverso cui l'elemento considerato passa durante la propria vita
- reagendo a eventi discreti (segnali, chiamate a funzionalità...).

## Statechart Diagram: diagrammi di stato

---

- Si possono pensare come “il contrario” degli Activity Diagram
- Enfasi posta sugli **stati** e non sulle **azioni**

Ing. SW: il ruolo di UML - 189

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Statechart Diagram: simboli base

---

 Inizio di un diagramma di stato

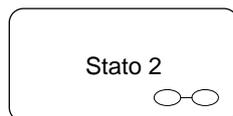
 Termine di un diagramma di stato



Singolo stato generico con nome “Stato 1”



Connessione fra stati



Stato scomponibile in un ulteriore diagramma di stati

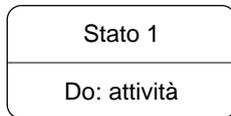
Nome del diagramma di stato (in fondo)

Ing. SW: il ruolo di UML - 190

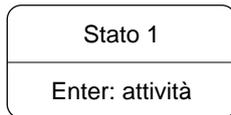
Giulio Destri - © AreaSP for Univ. Parma, 2006

## Statechart Diagram: simboli base

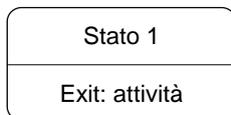
---



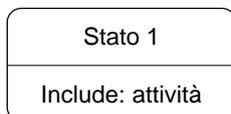
Stato con attività in corso per tutta la sua durata



Stato con attività che accade all'ingresso in esso



Stato con attività che accade all'uscita da esso



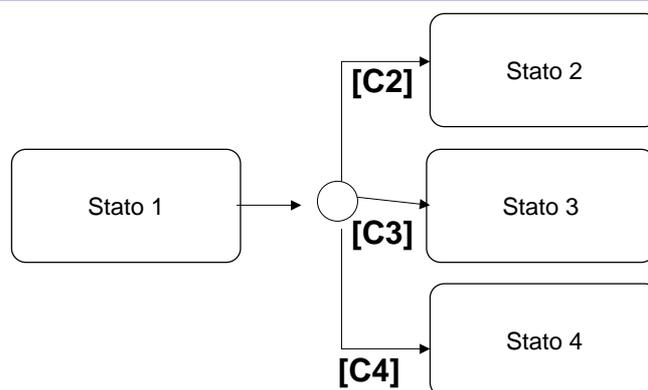
Stato che include un'attività compiuta su un altro diagramma

Ing. SW: il ruolo di UML - 191

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di SD: punto di scelta dinamica

---



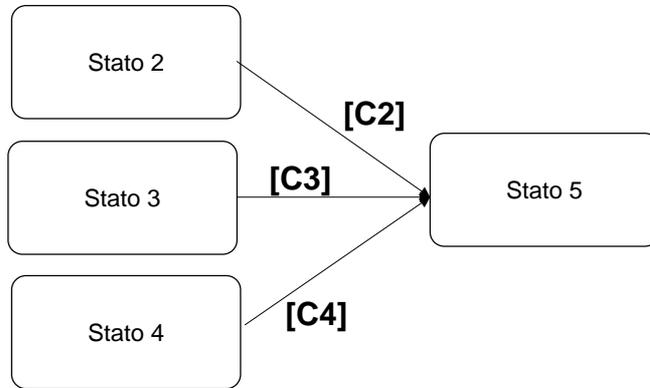
In base alle condizioni si passa ad uno degli stati.  
Forma equivalente: le frecce partono direttamente da Stato 1

Ing. SW: il ruolo di UML - 192

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di SD: punto di giunzione

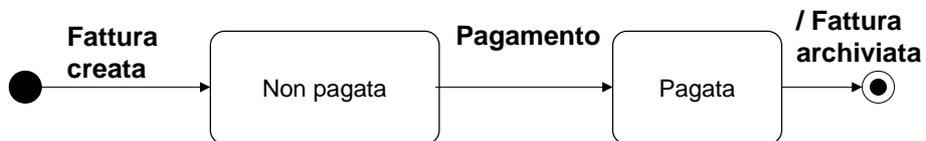
---



Il successivo di tutti e 3 gli stati, seguendo le condizioni, è Stato 5

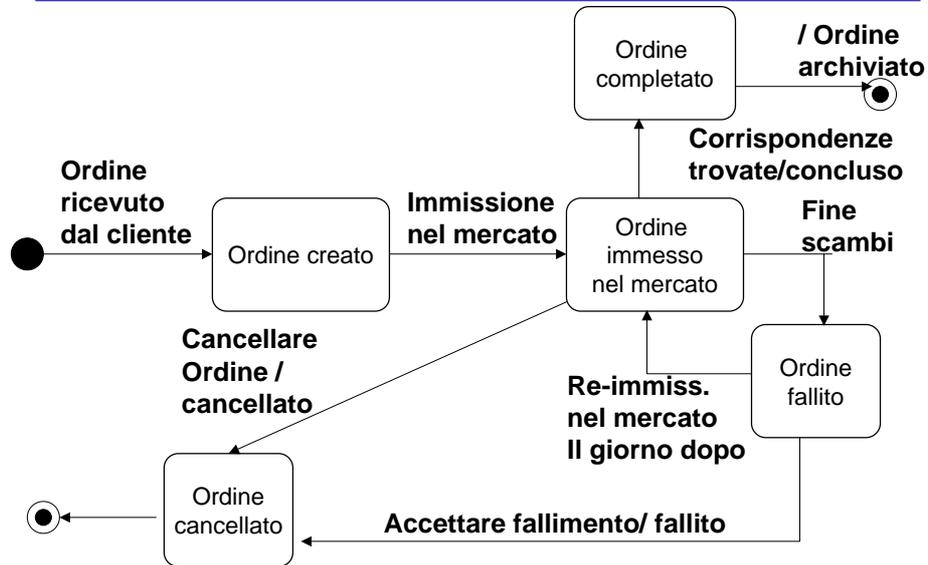
## Statechart Diagram: esempio

---



I rettangoli rappresentano uno stato per le entità in gioco

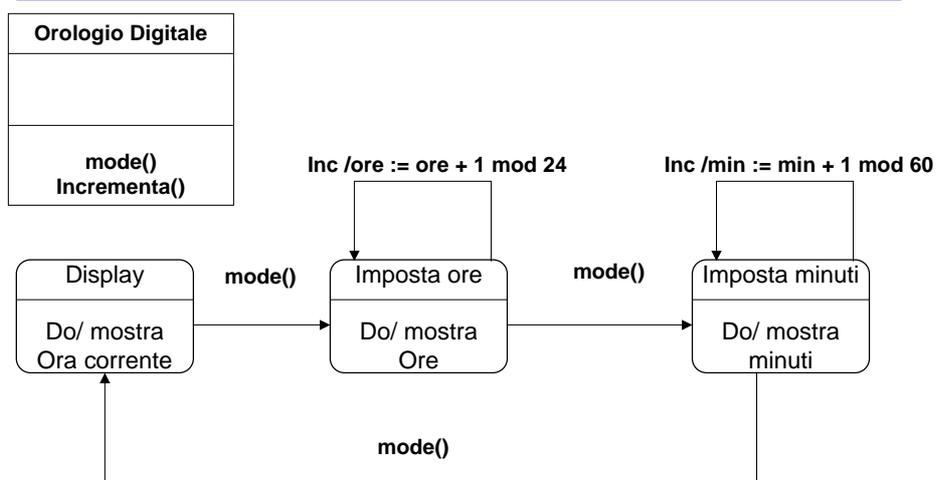
## Statechart Diagram: terminazione a due possibilità



Ing. SW: il ruolo di UML - 195

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Statechart Diagram: inserimento di oggetti

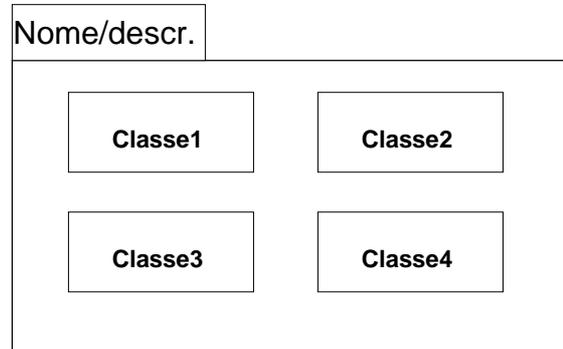


Ing. SW: il ruolo di UML - 196

Giulio Destri - © AreaSP for Univ. Parma, 2006

## I Package di UML

---



**Un package è un raggruppamento generale di elementi correlati fra loro da un legame logico che il modellatore ritiene importante. I package possono essere inseriti in tutti i diagrammi UML**

Ing. SW: il ruolo di UML - 197

Giulio Destri - © AreaSP for Univ. Parma, 2006



### **I diagrammi di analisi**

- **Class Diagram**
- **Activity Diagram**
- **Statechart Diagram**
- **Use Case Diagram**
- **Sequence Diagram**
- **Collaboration Diagram**

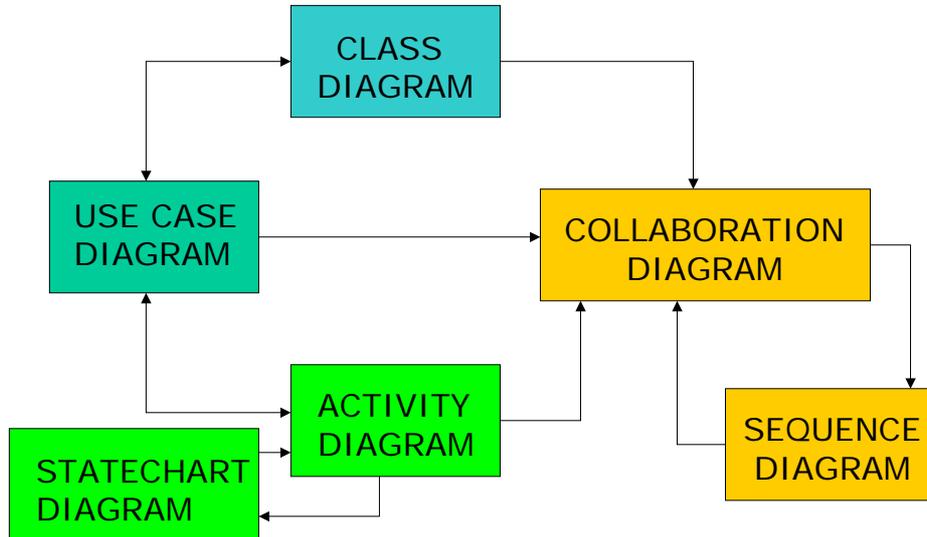
**Come sono imparentati fra loro?**

Ing. SW: il ruolo di UML - 198

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Il legame fra i diagrammi UML

---

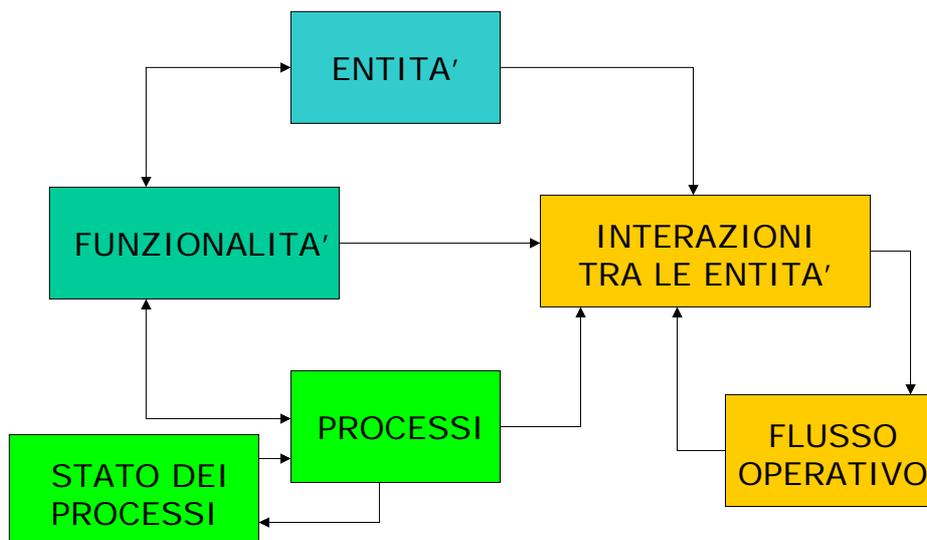


Ing. SW: il ruolo di UML - 199

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Il legame fra i diagrammi UML

---



Ing. SW: il ruolo di UML - 200

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Diagrammi di implementazione:

- **Component Diagram**
- **Deployment Diagram**

## Component Diagram: i componenti

---

- Un componente rappresenta un pezzo “fisico” dell’implementazione di un sistema
- La granularità della suddivisione del sistema dipende fortemente
  - dal contesto e
  - dal livello di astrazione in cui ci si pone

## Component Diagram: i componenti - 2

---

Attraverso l'uso degli stereotipi vengono classificati diversi componenti (lo standard UML ne definisce 5)

- programma eseguibile (es. .exe):  
<<executable>>
- libreria statica (es. .h) o dinamica (es. .dll): <<library>>
- file di codice sorgente o dati (es. .cpp):  
<<file>>
- documento (es. .htm): <<document>>
- tabella di database: <<table>>

## Component Diagram: i componenti - 3

---

Un componente ha un **nome** e una **locazione**

- è mostrato, tipicamente, con il solo nome
- per le classi, è possibile inserire compartimenti riportanti altri dettagli (analogia col class diagram)
- è possibile indicare le relazioni tra componenti e classi e/o interfacce che essi realizzano

## Component Diagram: i componenti - 4

---

- I componenti (come a livello logico le classi o altri elementi dei diagrammi) possono essere raggruppati in package
- Se i componenti sono file, i package sono cartelle/ directory

## Component Diagram: il diagramma dei componenti

---

- Rappresenta l'implementazione del sistema, attraverso la visione dei suoi elementi costitutivi
- Un componente rappresenta un pezzo "fisico" dell'implementazione di un sistema
- La suddivisione dipende dalla granularità del modello

## Component Diagram: il diagramma dei componenti - 2

---

- Definisce le relazioni fra i componenti software che realizzano l'applicazione
  - sorgenti, binari, eseguibili, ...
- Può operare a livelli diversi, ad esempio:
  - nella fase di sviluppo di un'applicazione può esprimere la struttura di un makefile
  - nella fase di installazione può indicare le dipendenze da librerie, file di configurazione...

## Component Diagram: il diagramma dei componenti - 3

---

- Può rappresentare parte della specifica architetturale
- Evidenzia l'organizzazione e le dipendenze esistenti tra componenti
- Evidenzia anche la distinzione fra i diversi componenti e le varie interfacce che essi offrono e usano
- Primo passo verso **Component Programming**

## Component Diagram: il diagramma dei componenti - 4

---

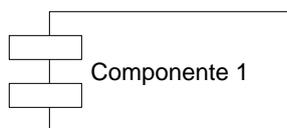
- Le interfacce rappresentano insiemi di operazioni (metodi) che definiscono un servizio
- Un componente può usare le interfacce di altri componenti
- Un componente può provvedere le proprie interfacce

Ing. SW: il ruolo di UML - 209

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Component Diagram: simboli base

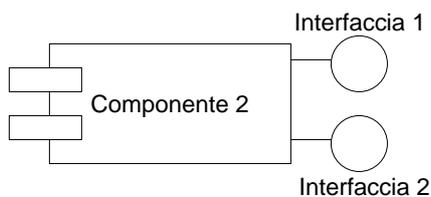
---



Singolo componente generico con nome "Componente 1"

<<tipo relazione>>  
→

Relazione fra componenti con <<tipo>>, di solito dipendenza



Componente con due interfacce, aventi nome interfaccia 1 e 2

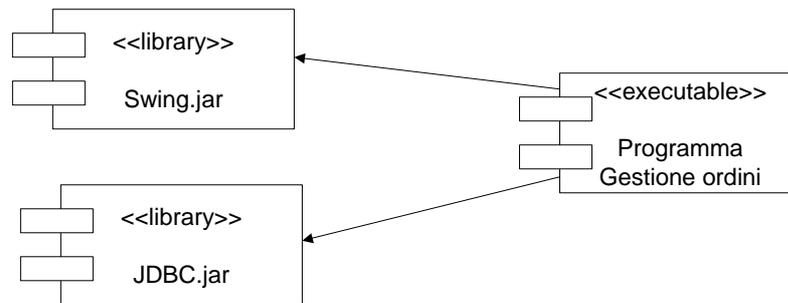
Nome del diagramma dei componenti (in fondo)

Ing. SW: il ruolo di UML - 210

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Esempio: eseguibile e librerie

---



Ing. SW: il ruolo di UML - 211

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Component Diagram: simboli alternativi

---

- Non esistono standard "assoluti" sui simboli da usare per i componenti
- Per esempio
  - Un file di testo può essere indicato con l'icona del blocco note
  - Un documento con l'icona del documento
  - Una libreria con un quadrato con ingranaggi

Ing. SW: il ruolo di UML - 212

Giulio Destri - © AreaSP for Univ. Parma, 2006

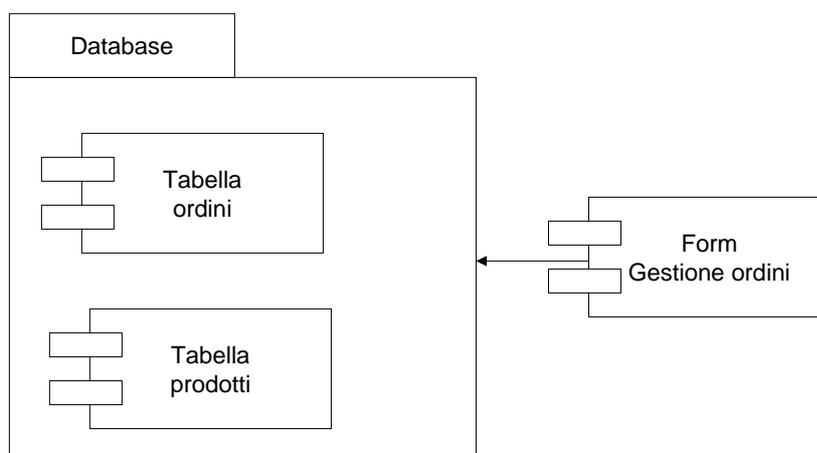
## Component Diagram e altri diagrammi

---

- Le corrispondenze fra gli elementi del Component Diagram e gli elementi di altri diagrammi dipendono dalle scelte fatte nella granularità del problema
- Ad esempio un singolo componente può corrispondere ad una classe, ad un intero gruppo di classi o un'intera applicazione
- I package possono essere di aiuto per chiarificare le cose

### Esempio: il package come "contenitore"

---



## Categorie di componenti: deployment

---

I **deployment components** sono i componenti necessari e sufficienti per formare un sistema eseguibile

- Eseguibili (exe, com...)
- Librerie (DLL, .so, .jar...)
- File di configurazione propri del programma

## Categorie di componenti: work product

---

- I **work product components** sono componenti che non partecipano direttamente nel sistema eseguibile ma che sono frutto del lavoro fatto per creare il sistema eseguibile
- Essenzialmente si possono pensare come il "residuo" del processo di sviluppo
  - File sorgenti
  - File di configurazione dei progetti (project file o XML file di ants o nants...)
  - Data file usati per la creazione di deployment

## Categorie di componenti: execution

---

- Gli **execution component** sono componenti creati per permettere il funzionamento del sistema eseguibile
- Dipendono fortemente dall'ambiente (sistema operativo e "infrastrutture") entro cui il sistema eseguibile opera
  - COM+ object
  - JCL
  - Interfaccia verso MQseries

## Deployment Diagram

---

- Rappresenta la distribuzione dei componenti di un sistema eseguibile fra le risorse disponibili
- La granularità della suddivisione del sistema dipende fortemente
  - dal contesto e
  - dal livello di astrazione in cui ci si pone

## Deployment Diagram - 2

---

- Il diagramma può essere riferito a risorse hardware (es. server, nodi di calcolo)
- Oppure a macroelementi software (es. mappatura dei componenti logici entro i processi)
- O anche a sistemi informatici completi (es. sistemi informatici distribuiti), in congiunzione con collaboration diagram

## Deployment Diagram - 3

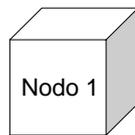
---

In pratica quindi

- Consente di rappresentare, a diversi livelli di dettaglio, l'architettura fisica (hardware e software) di un sistema
- Ma anche di evidenziare la configurazione (run-time o no)
  - dei singoli nodi elaborativi
  - dei singoli componenti software (processi, oggetti...)

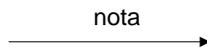
## Deployment Diagram: simboli base

---



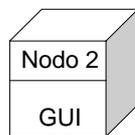
Singolo nodo (semplice)  
generico con nome "Nodo 1"

<<tipo comunicazione>>



Comunicazione fra nodi con <<tipo>>, di solito indicante il protocollo

Relazione/nota esplicativa



Singolo nodo (esteso)  
generico con nome "Nodo 2"  
e indicazione dei componenti

Nome del diagramma di dislocazione (in fondo)

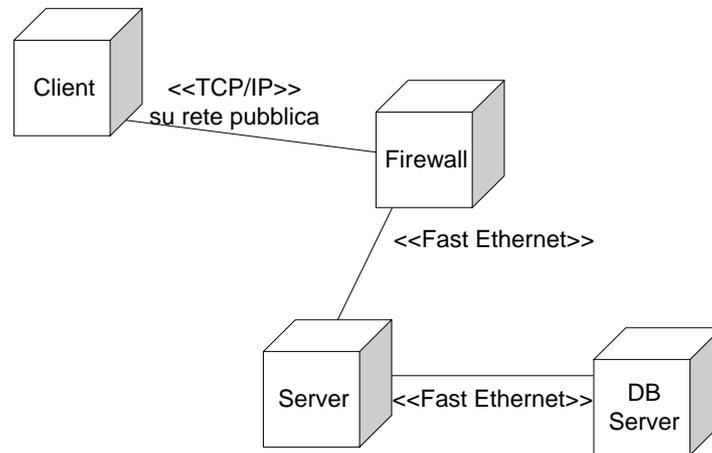
## Deployment Diagram: simboli alternativi

---

- Anche per il deployment sono usati spesso altri simboli
- Per esempio
  - Un DB server può essere indicato col cilindro
  - Un server con un computer stilizzato
  - Una postazione client con il monitor stilizzato

## Deployment Diagram: esempio

---



Esempio di semplice sistema client/server

Ing. SW: il ruolo di UML - 223

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Deployment e Component Diagram

---

- Alle volte per completezza si possono inserire componenti anche nei deployment diagram
- In generale
  - componenti sono "entità" che partecipano nell'esecuzione di un sistema
  - nodi sono "entità" che eseguono componenti
  - componenti rappresentano il packaging fisico di altri elementi logici
  - nodi rappresentano l'allocazione fisica di componenti

Ing. SW: il ruolo di UML - 224

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

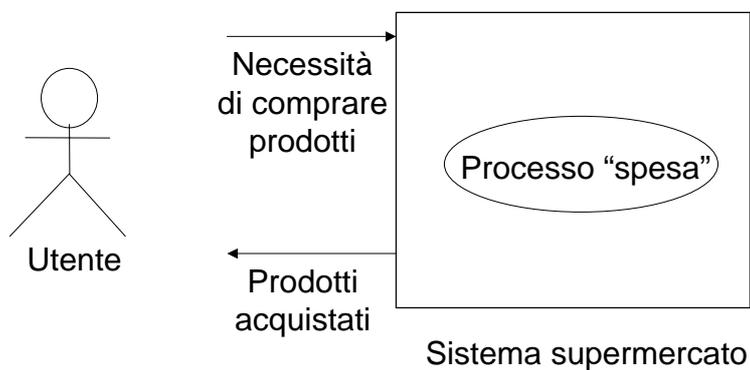
- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

Ing. SW: il ruolo di UML - 225

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un processo del mondo reale: la spesa

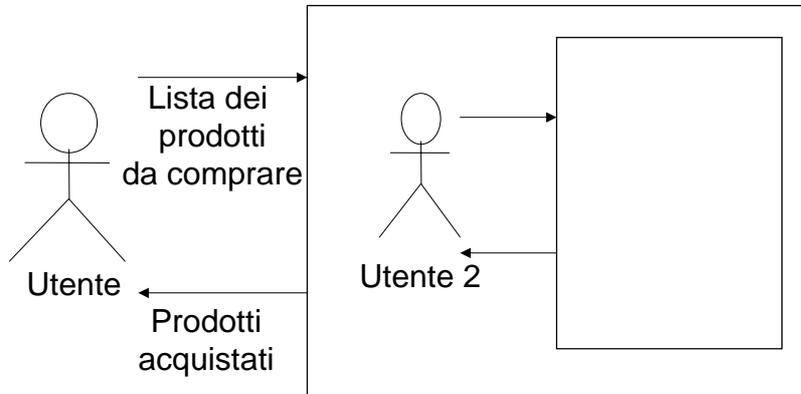
---



Ing. SW: il ruolo di UML - 226

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Quali sono i limiti di un sistema?

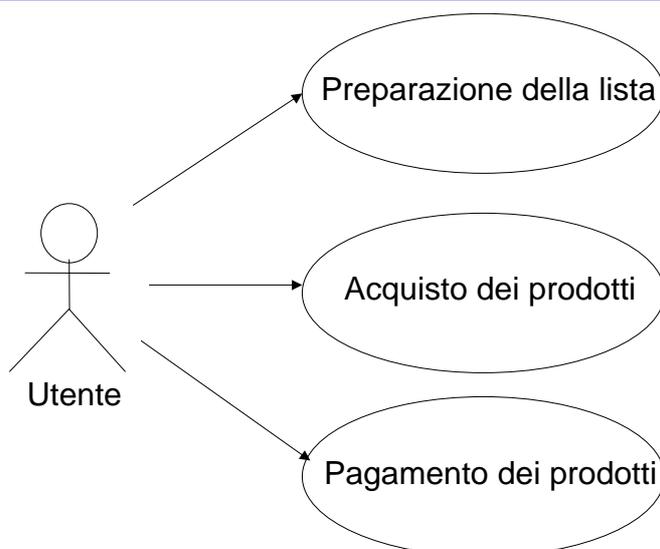


Sistema: utente 2 + negozio  
Processo: "fai la spesa"

Ing. SW: il ruolo di UML - 227

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le funzionalità della spesa: use case

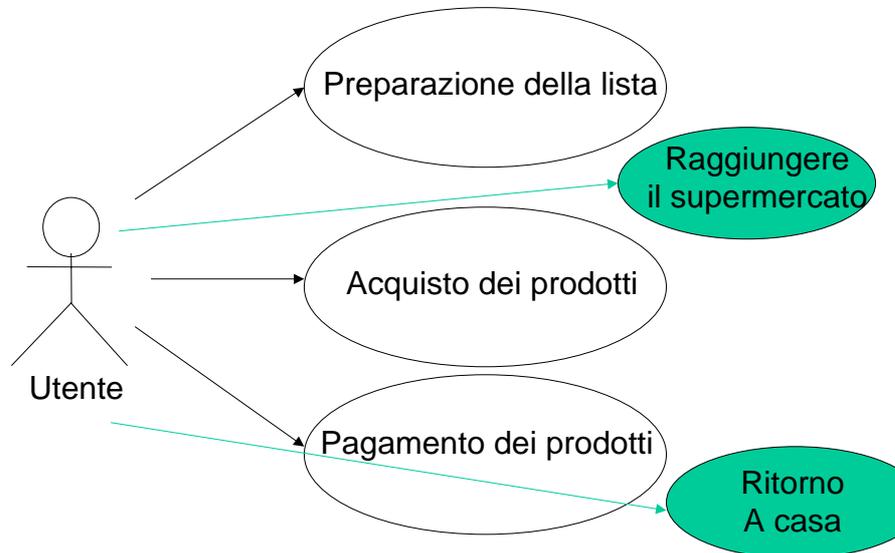


Ing. SW: il ruolo di UML - 228

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le funzionalità della spesa: use case

---

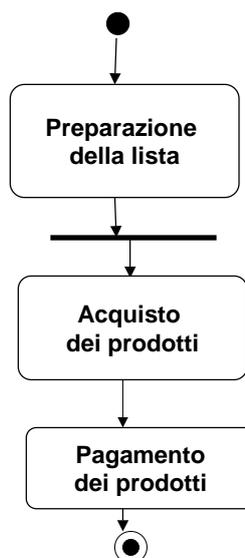


Ing. SW: il ruolo di UML - 229

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Spesa: Diagramma di attività 1

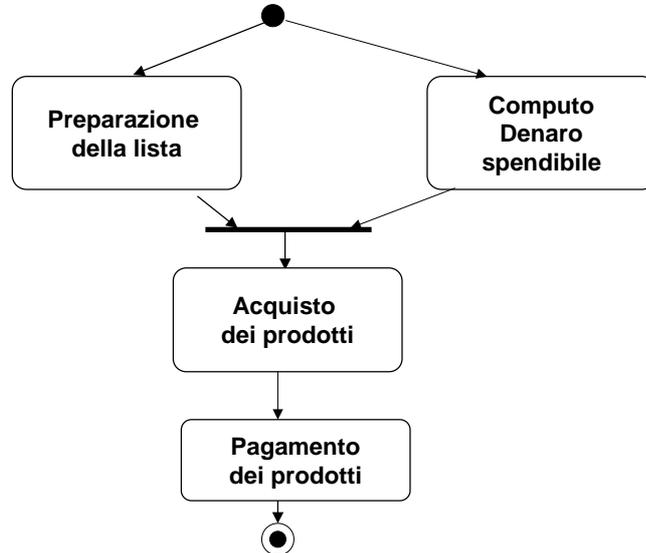
---



Ing. SW: il ruolo di UML - 230

Giulio Destri - © AreaSP for Univ. Parma, 2006

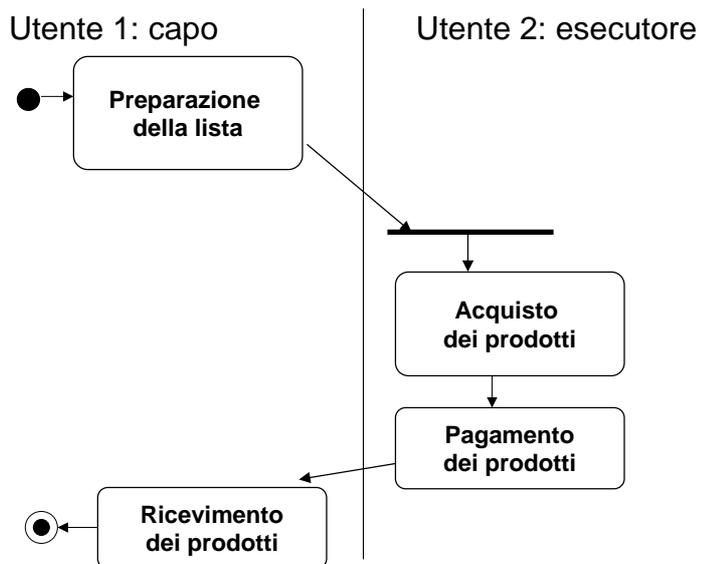
## Spesa: Diagramma di attività 1



Ing. SW: il ruolo di UML - 231

Giulio Destri - © AreaSP for Univ. Parma, 2006

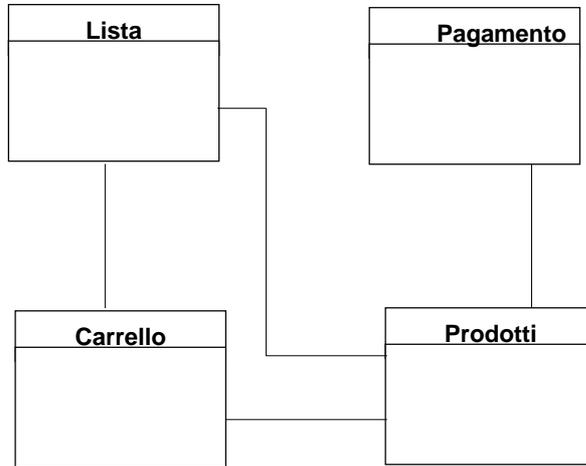
## Spesa: Diagramma di attività 2



Ing. SW: il ruolo di UML - 232

Giulio Destri - © AreaSP for Univ. Parma, 2006

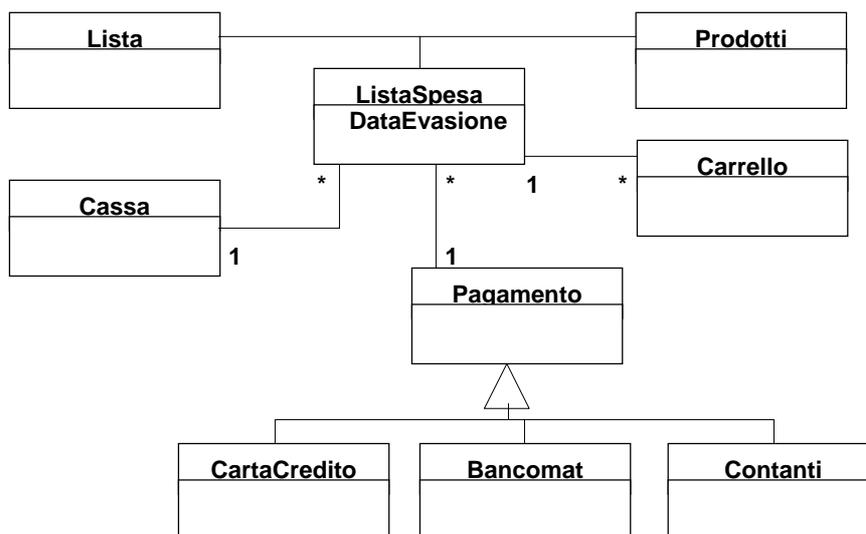
## Spesa: cercare le classi



Ing. SW: il ruolo di UML - 233

Giulio Destri - © AreaSP for Univ. Parma, 2006

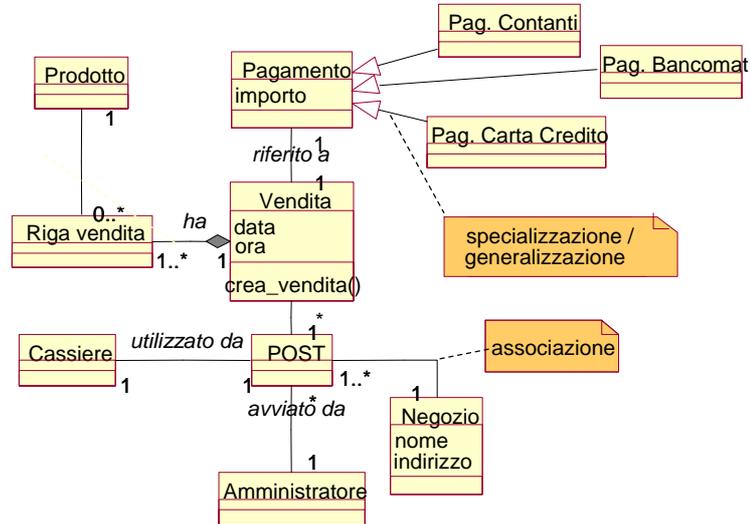
## Spesa: Diagramma delle classi



Ing. SW: il ruolo di UML - 234

Giulio Destri - © AreaSP for Univ. Parma, 2006

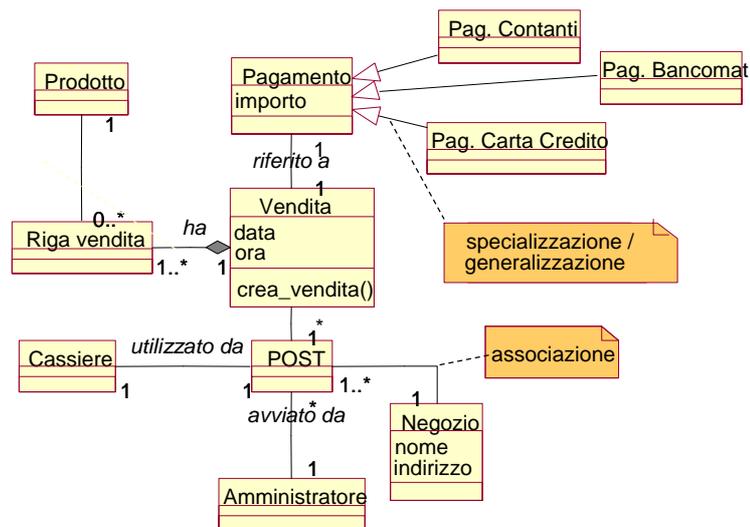
## Spesa: Class Diagram completo



Ing. SW: il ruolo di UML - 235

Giulio Destri - © AreaSP for Univ. Parma, 2006

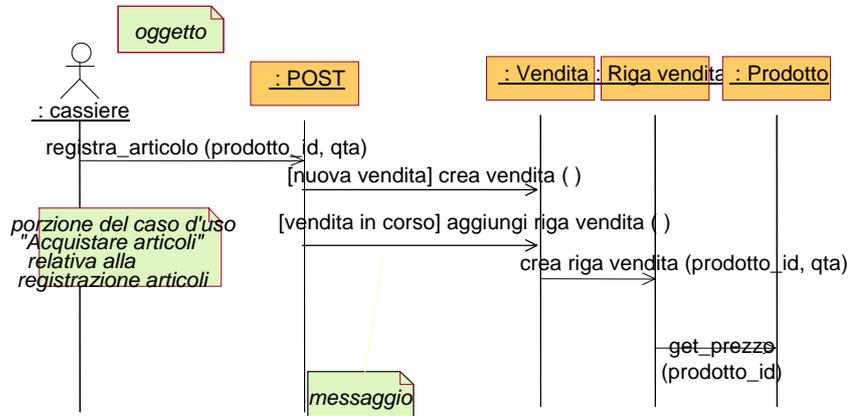
## Spesa: Class Diagram completo



Ing. SW: il ruolo di UML - 236

Giulio Destri - © AreaSP for Univ. Parma, 2006

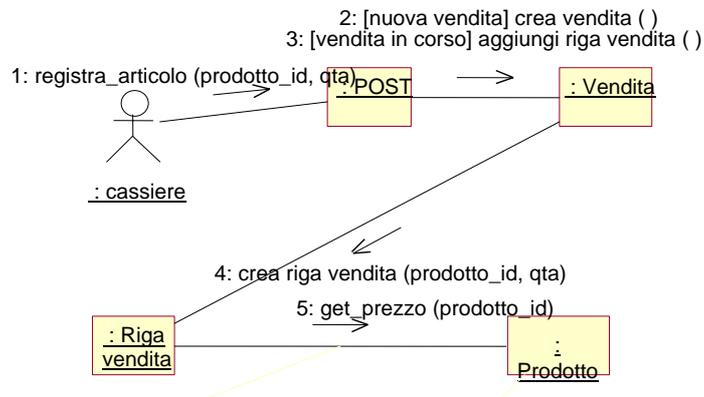
## Spesa: diagramma di sequenza



Ing. SW: il ruolo di UML - 237

Giulio Destri - © AreaSP for Univ. Parma, 2006

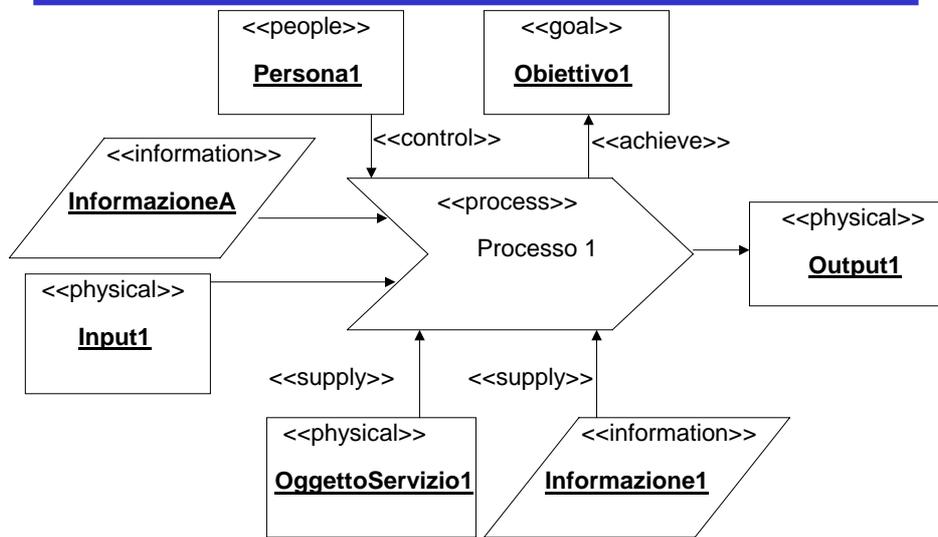
## Spesa: collaborazione



Ing. SW: il ruolo di UML - 238

Giulio Destri - © AreaSP for Univ. Parma, 2006

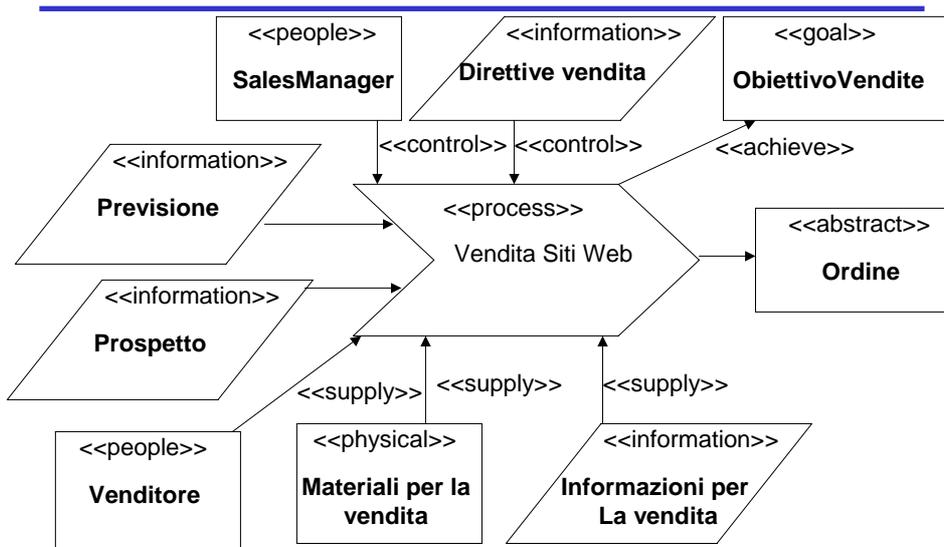
## Activity Diagram: il processo generico



Ing. SW: il ruolo di UML - 239

Giulio Destri - © AreaSP for Univ. Parma, 2006

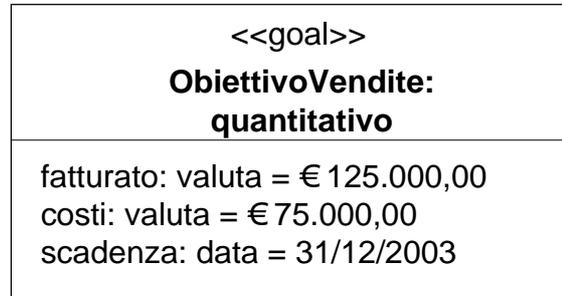
## Esempio di AD: vendita di siti Web



Ing. SW: il ruolo di UML - 240

Giulio Destri - © AreaSP for Univ. Parma, 2006

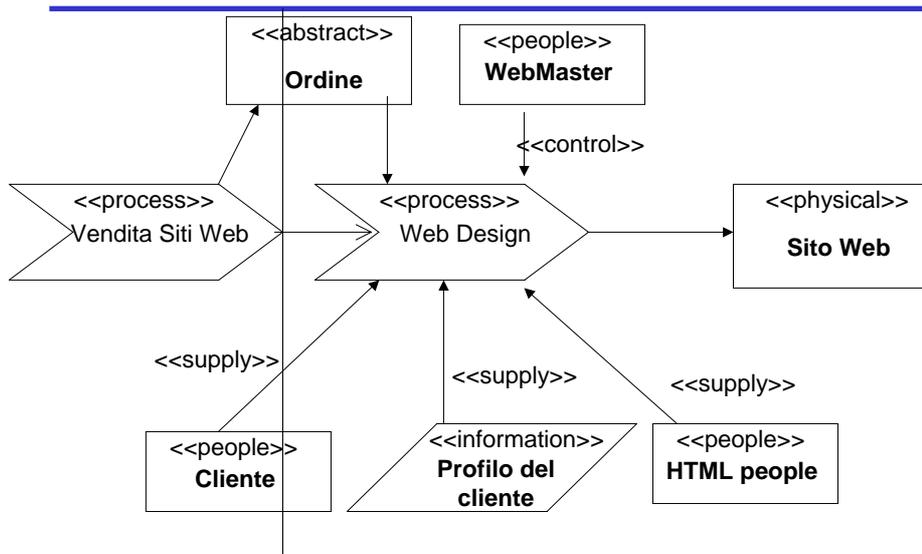
## Un esempio di goal: l'obiettivo vendite espanso



Ing. SW: il ruolo di UML - 241

Giulio Destri - © AreaSP for Univ. Parma, 2006

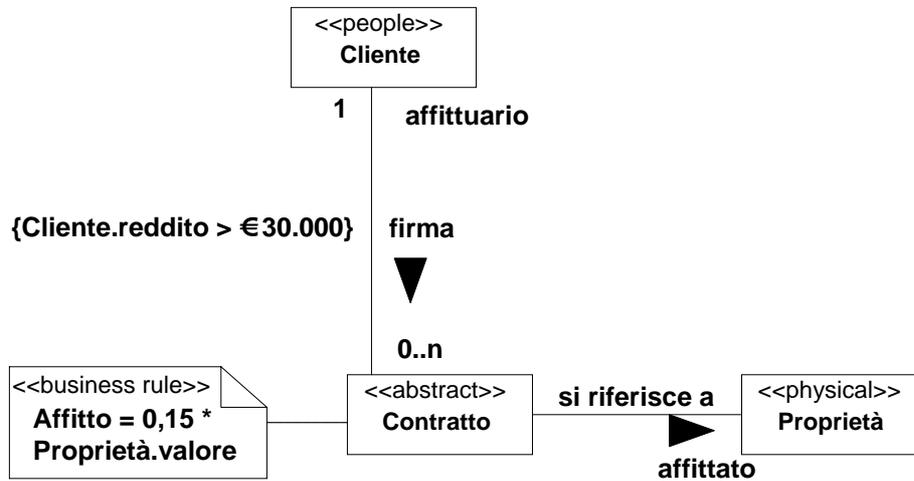
## Combinazione AD/swimlanes



Ing. SW: il ruolo di UML - 242

Giulio Destri - © AreaSP for Univ. Parma, 2006

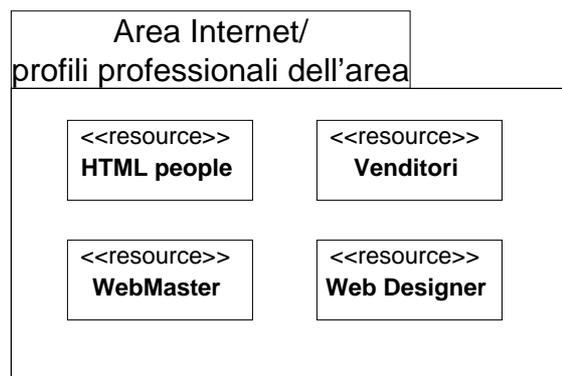
## Class Diagram per regole business



Ing. SW: il ruolo di UML - 243

Giulio Destri - © AreaSP for Univ. Parma, 2006

## I Package in UML for Business



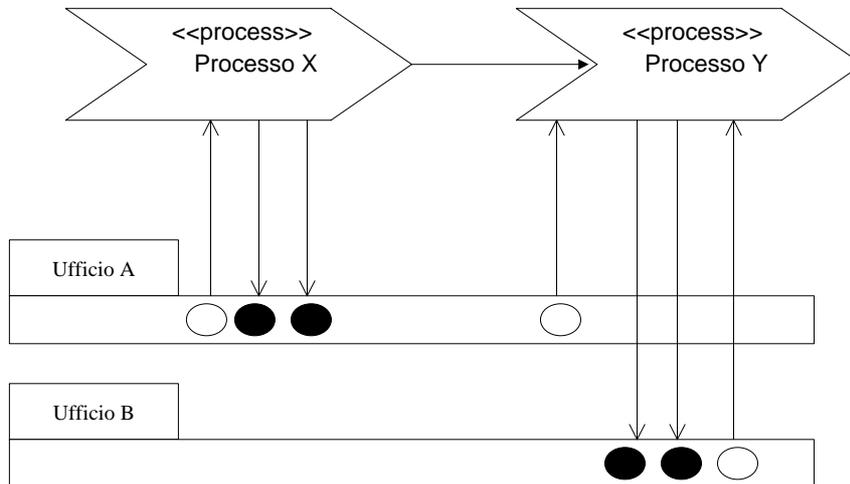
Un package rappresenta un raggruppamento di elementi correlati fra loro, per esempio i profili professionali che formano una Web agency

Ing. SW: il ruolo di UML - 244

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Assembly line

---



Ing. SW: il ruolo di UML - 245

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

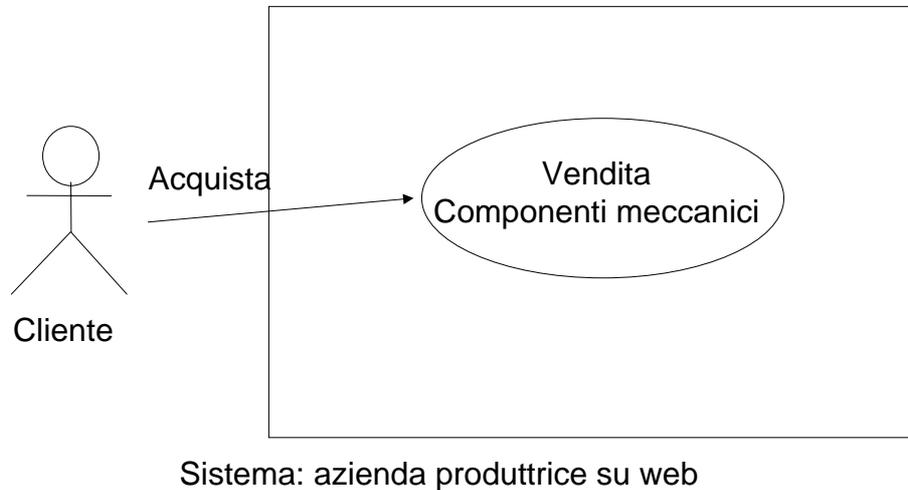
- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- [Case study](#)
- I pattern nell'analisi del business
- La visione dell'IT con oggetti e Web Services

Ing. SW: il ruolo di UML - 246

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Use Case: vendita pezzi meccanici via Web

---



Ing. SW: il ruolo di UML - 247

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Le entità coinvolte nel processo

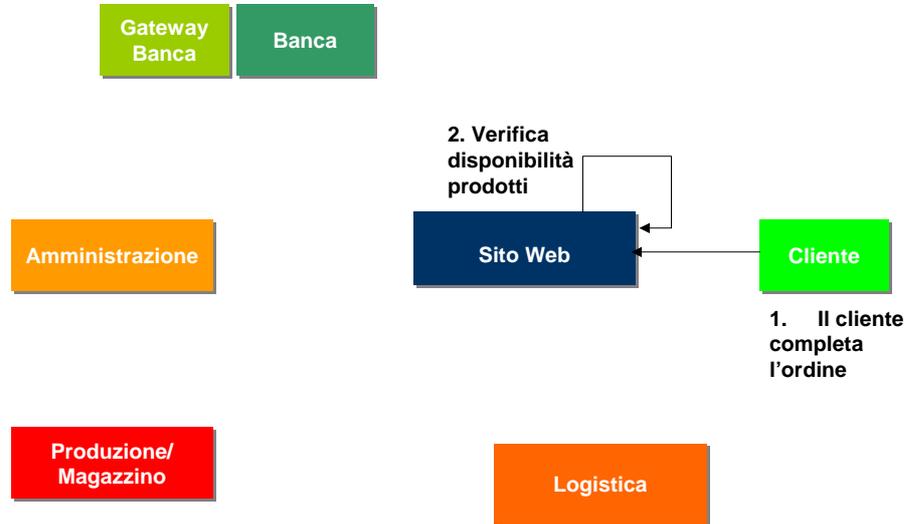
---

- Cliente
- Sito Web
- Amministrazione
- Produzione/Magazzino
- Logistica
- Banca

Ing. SW: il ruolo di UML - 248

Giulio Destri - © AreaSP for Univ. Parma, 2006

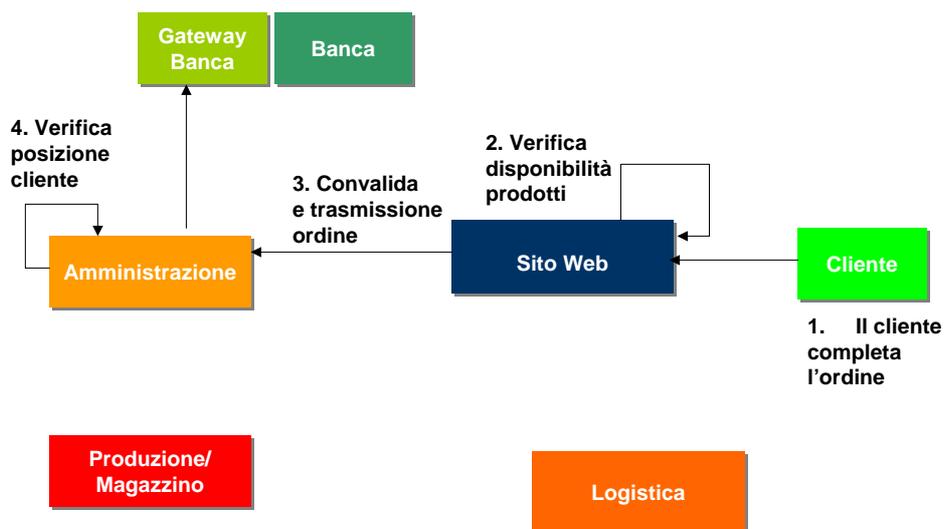
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 249

Giulio Destri - © AreaSP for Univ. Parma, 2006

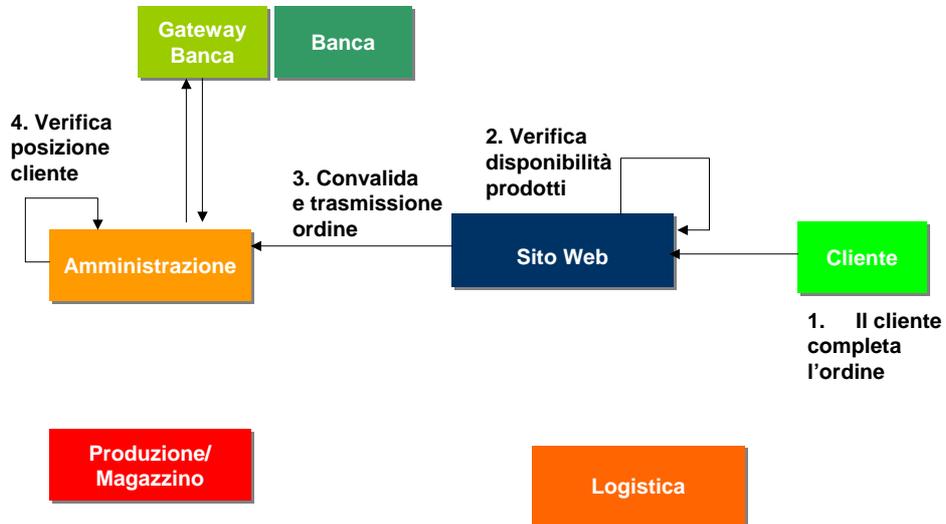
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 250

Giulio Destri - © AreaSP for Univ. Parma, 2006

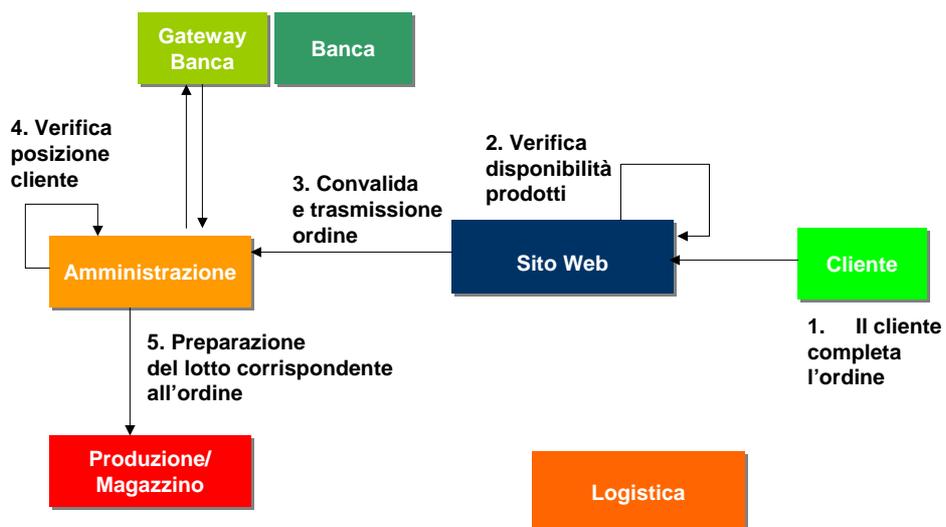
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 251

Giulio Destri - © AreaSP for Univ. Parma, 2006

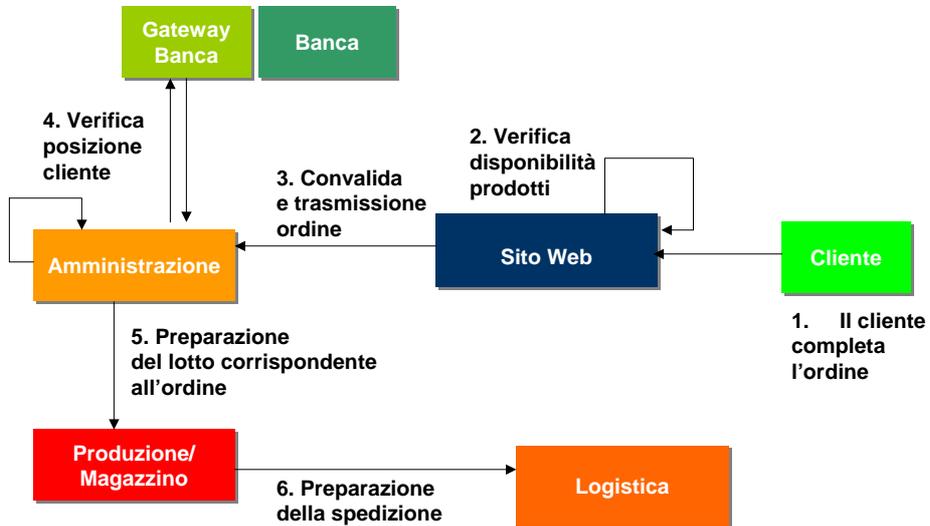
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 252

Giulio Destri - © AreaSP for Univ. Parma, 2006

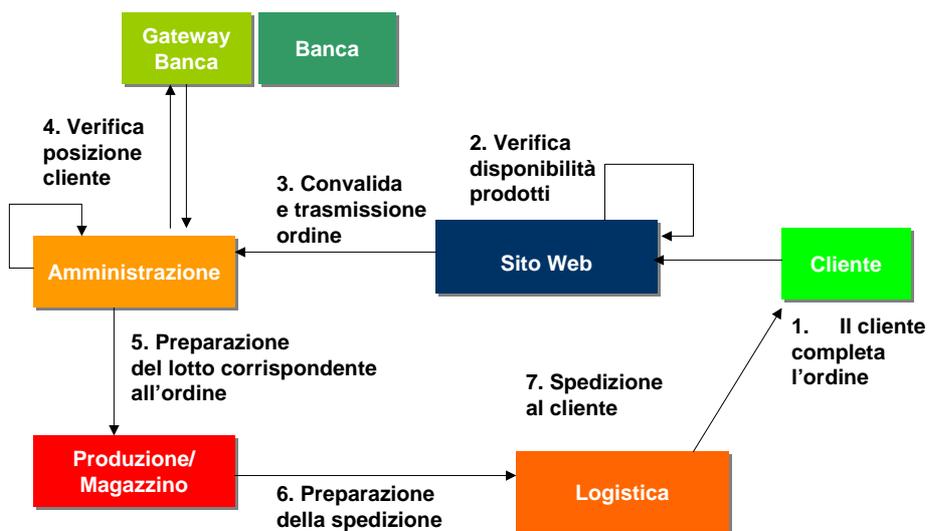
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 253

Giulio Destri - © AreaSP for Univ. Parma, 2006

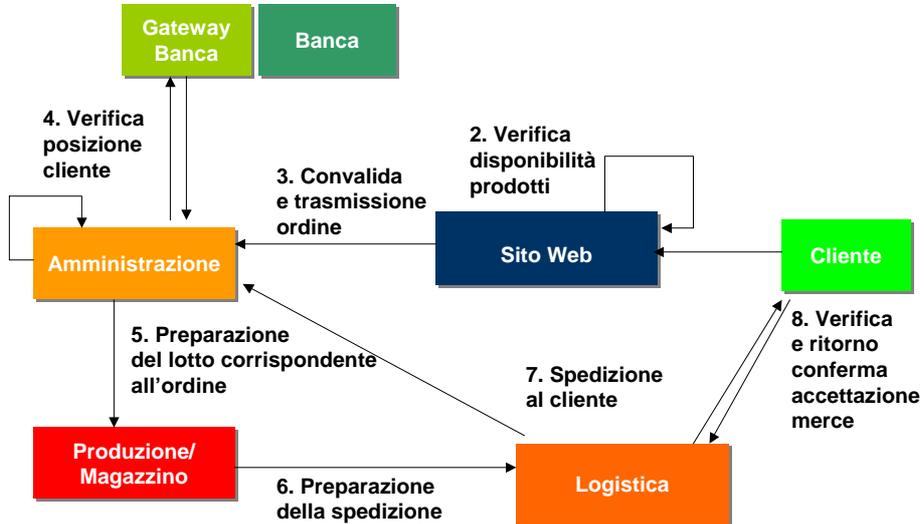
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 254

Giulio Destri - © AreaSP for Univ. Parma, 2006

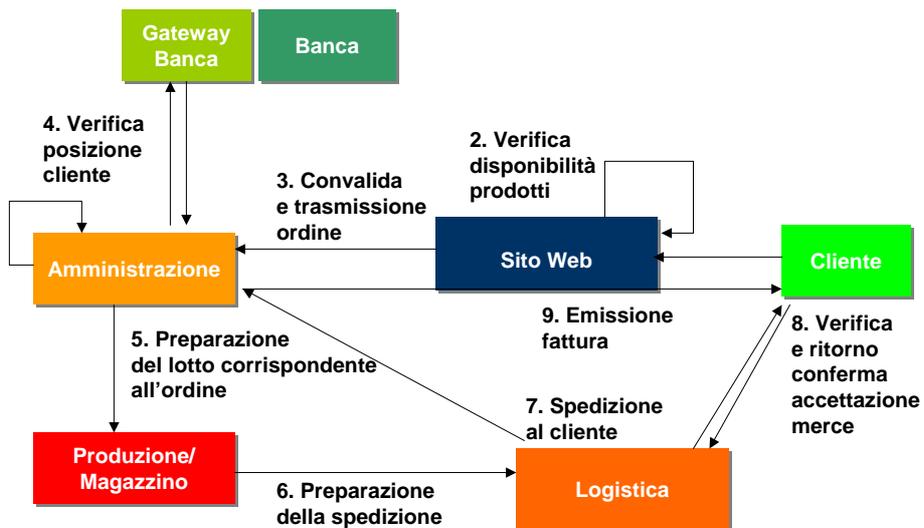
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 255

Giulio Destri - © AreaSP for Univ. Parma, 2006

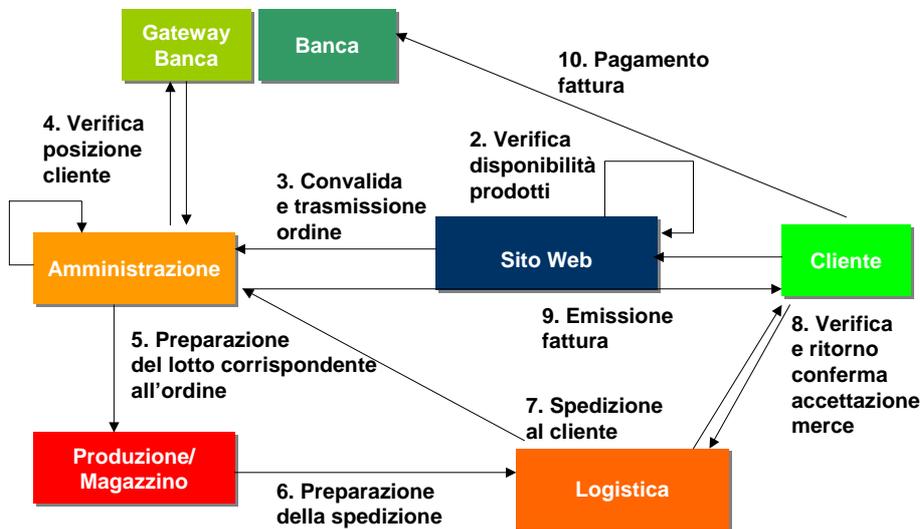
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 256

Giulio Destri - © AreaSP for Univ. Parma, 2006

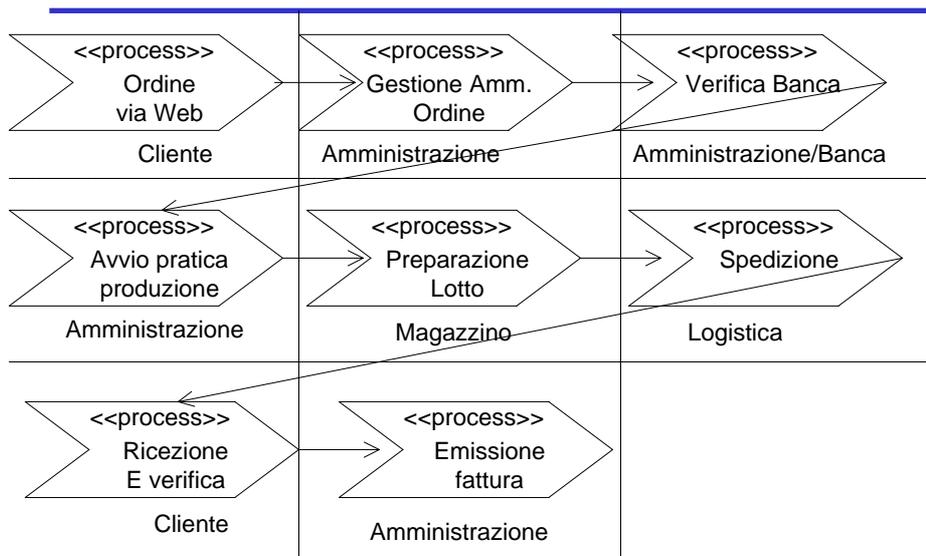
## Ciclo attivo: collaboration diagram



Ing. SW: il ruolo di UML - 257

Giulio Destri - © AreaSP for Univ. Parma, 2006

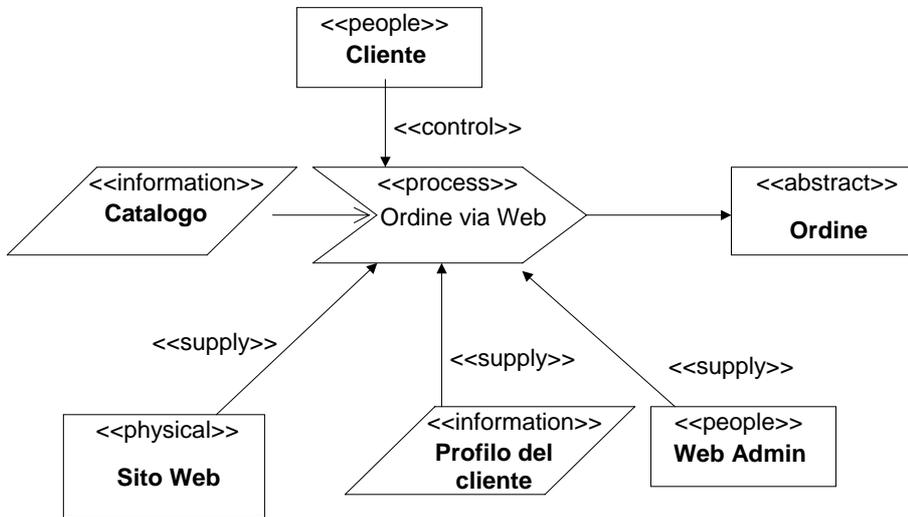
## Il ciclo attivo visto per processi



Ing. SW: il ruolo di UML - 258

Giulio Destri - © AreaSP for Univ. Parma, 2006

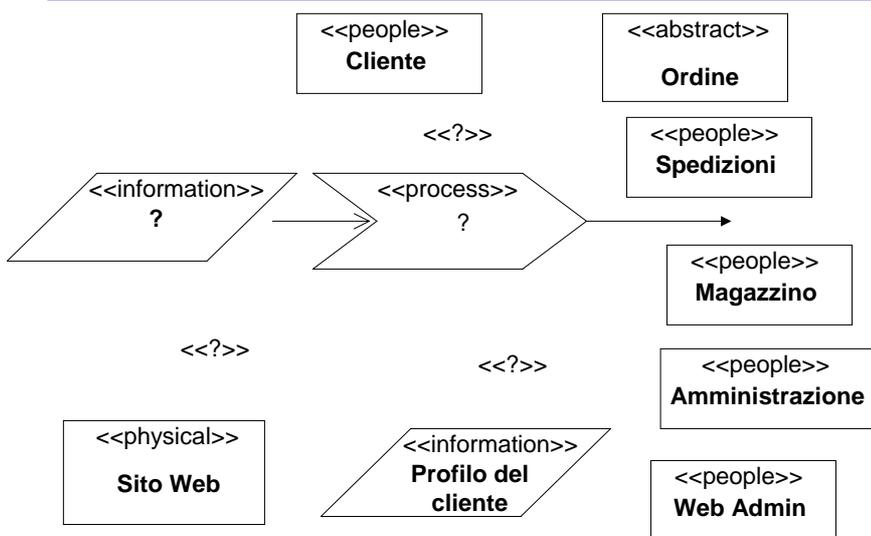
## Il ciclo attivo visto per processi



Ing. SW: il ruolo di UML - 259

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Altri processi nel ciclo attivo



Ing. SW: il ruolo di UML - 260

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- [I pattern nell'analisi del business](#)
- La visione dell'IT con oggetti e Web Services

Ing. SW: il ruolo di UML - 261

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Un Pattern...

---

- descrive un problema
- che ricorre in specifici contesti
- propone un generico, ma ben dimostrato, schema per la sua soluzione.

Ing. SW: il ruolo di UML - 262

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Quindi un Pattern è

---

- una soluzione
- ad un determinato problema
- in un definito contesto
  
- che, talvolta, può essere generalizzata ed adattata a molti contesti diversi

## E un Business Pattern

---

- Si riferisce a problemi di Business
- Tipicamente analizza situazioni di modellizzare e/o strutturare risorse business che comprendono documenti, organizzazione, informazioni

## Suddivisione dei Business Pattern

---

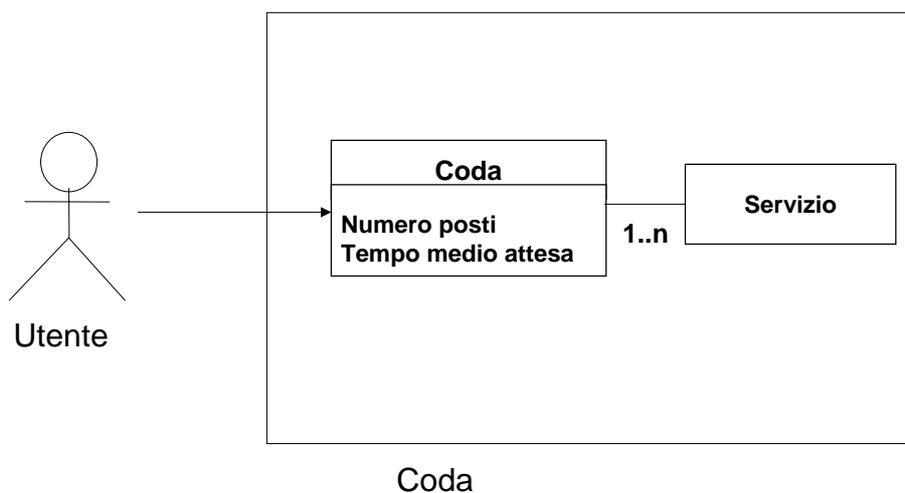
- Risorse e ruoli (Resource and Role patterns)
- Obiettivo (Goal patterns)
- Processo (Process patterns)

Ing. SW: il ruolo di UML - 265

Giulio Destri - © AreaSP for Univ. Parma, 2006

## La coda: pattern di processo

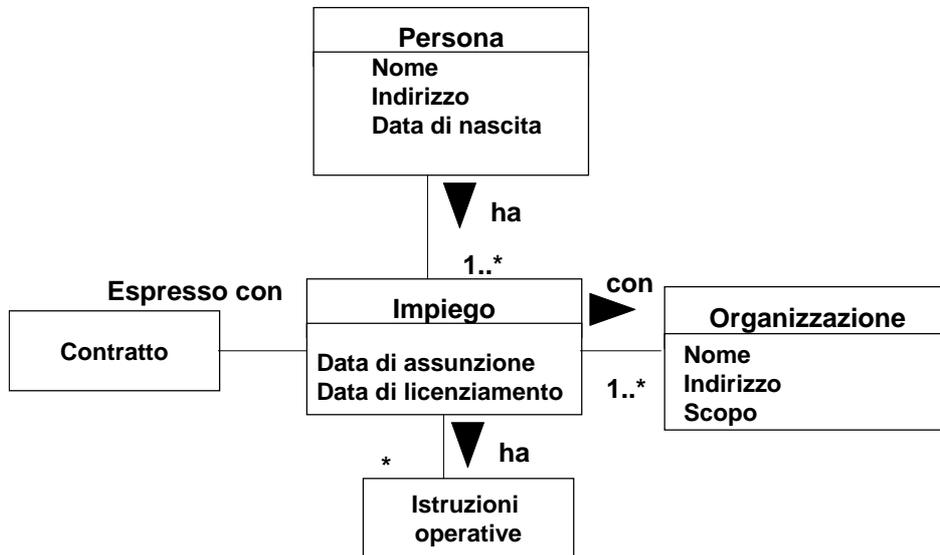
---



Ing. SW: il ruolo di UML - 266

Giulio Destri - © AreaSP for Univ. Parma, 2006

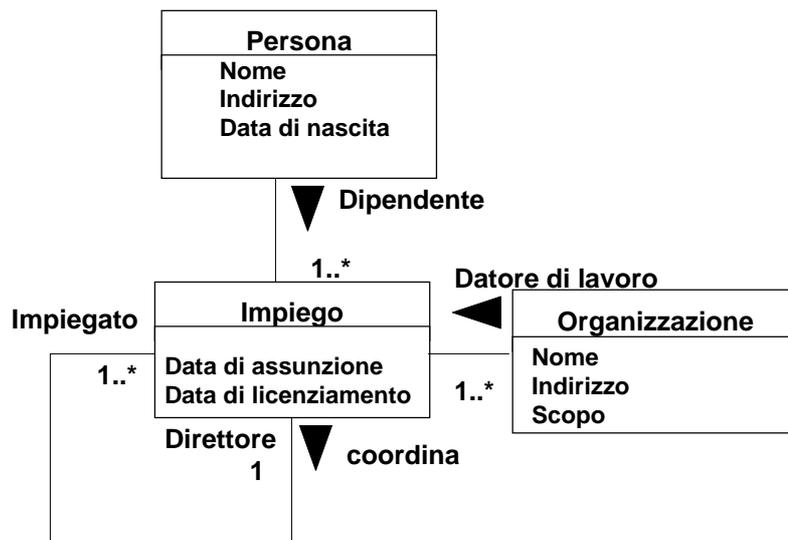
## L'impiego: pattern di ruolo



Ing. SW: il ruolo di UML - 267

Giulio Destri - © AreaSP for Univ. Parma, 2006

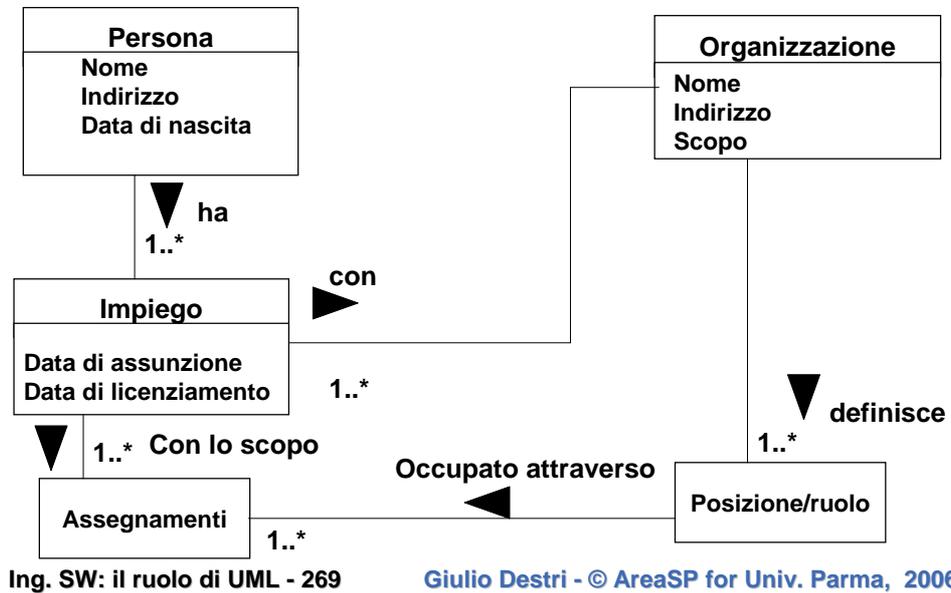
## L'impiego: pattern di ruolo



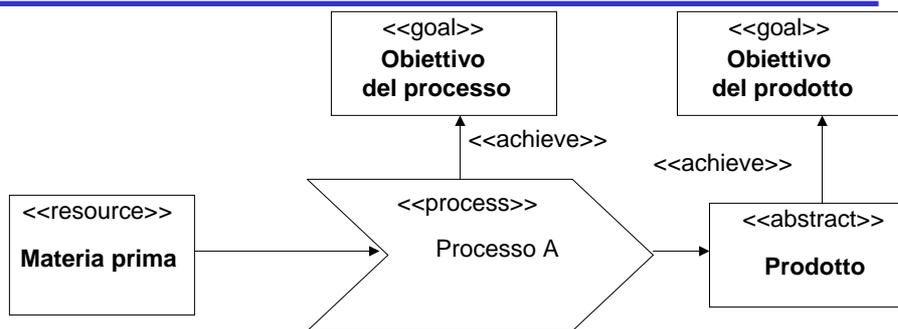
Ing. SW: il ruolo di UML - 268

Giulio Destri - © AreaSP for Univ. Parma, 2006

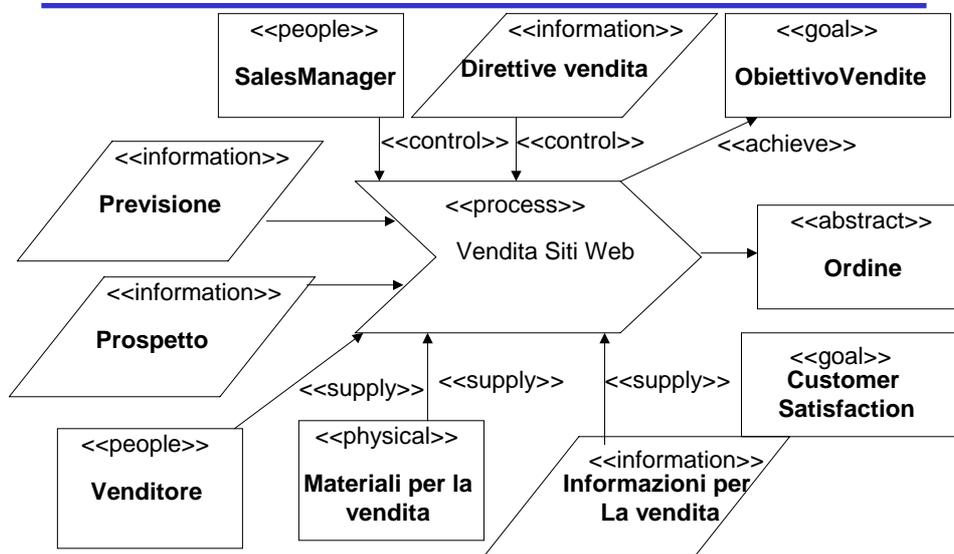
## L'impiego: pattern di ruolo



## Business goal allocation: goal pattern



## I goal nella vendita di siti Web



Ing. SW: il ruolo di UML - 271

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

- Il contesto: la realtà e la sua modellazione
- I sistemi informativi
- L'evoluzione di UML
- Dall'IT modeling al business modeling
- Come si usano i diagrammi
- Esempi di processi e attività business
- Case study
- I pattern nell'analisi del business
- [La visione dell'IT con oggetti e Web Services](#)

Ing. SW: il ruolo di UML - 272

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Uso di UML: il mercato IT

---

- Avvento dei linguaggi Object-Oriented
- Mosse degli attori principali dell'IT
  - IBM acquista Rational (Ottobre 2002)
  - Microsoft acquista Navision e Great Plain (Novembre 2002)
  - Avvento dei Web Services per l'integrazione (2001-2002)
  - IBM, Microsoft e Bea annunciano congiuntamente lo standard comune BPEL4WS (estate 2002)

## Uso di UML: il mercato IT

---

- I sistemi informativi OO-based (spot televisivi di Microsoft e IBM)
- "Interconnessione totale" entro l'azienda
- Flessibilità ed economicità delle strutture IT
- Riorganizzazione rapida ed efficace dei processi informativi (e quindi anche di quelli produttivi)

## Uso di UML: il mercato IT

---

- Individuazione degli "oggetti esecutivi" che compiono le varie fasi di cui ogni processo si compone
- Riduzione a "fattor comune" degli oggetti medesimi

## Uso di UML: il mercato IT

---

- Oggetti che possono corrispondere alla "storica" suddivisione dell'azienda in unità funzionali (vendite, marketing, amministrazione...)
- O anche a unità funzionali "più piccole" entro l'azienda, in funzione della scomposizione fatta e della conseguente "granularità" della ingegnerizzazione del processo

## Uso di UML: il mercato IT

---

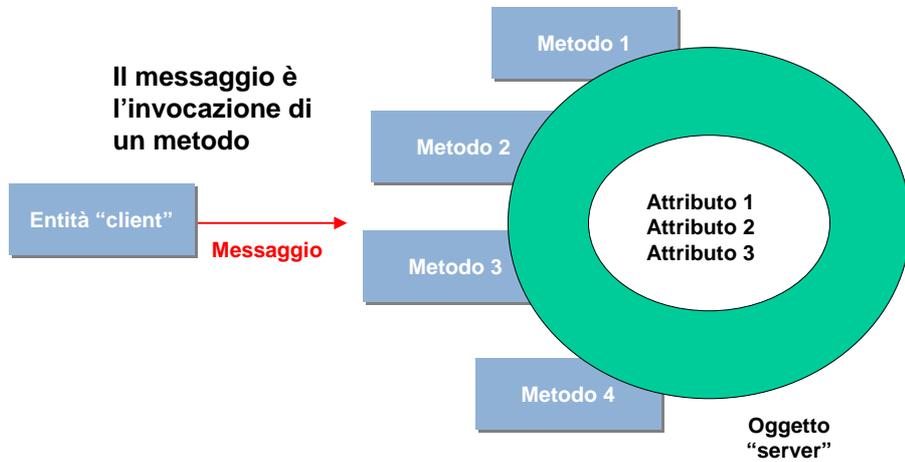
- La tecnologia dei Web Services può condurre alla realizzazione della infrastruttura IT per una organizzazione di questo tipo entro l'azienda:
  - "Sovrapponendosi" all'esistente
  - Senza stravolgere l'esistente

## Web Services (Servizi Web)

---

- Server Web
- Offrono possibilità di interazione completa (RPC) tramite SOAP
- Offrono ogni tipo di servizio

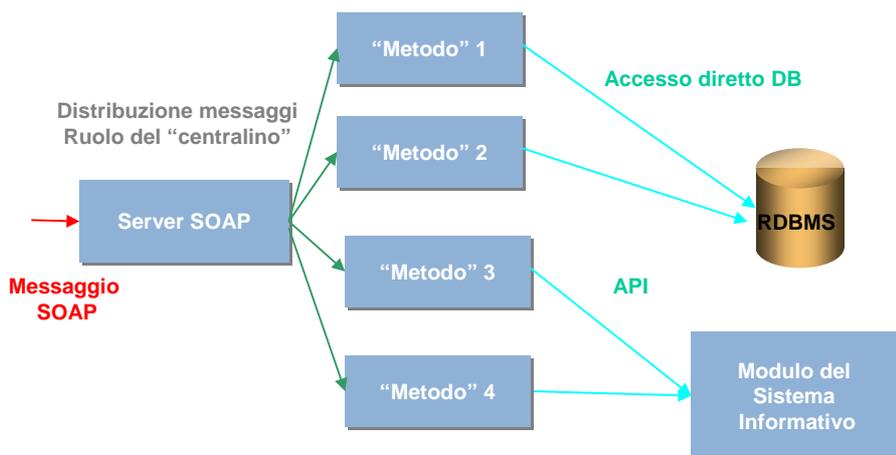
## Oggetto



Ing. SW: il ruolo di UML - 279

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Oggetto "Web Service"

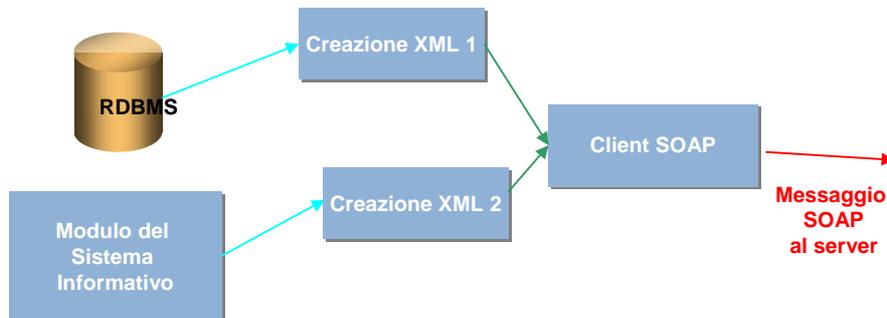


Ing. SW: il ruolo di UML - 280

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Oggetto "Web service client"

---



Ing. SW: il ruolo di UML - 281

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Bibliografia: UML

---

M. Fowler  
UML distilled, 3rd ed  
Ed. Prentice Hall, 2003

L. Vetti Tagliati  
UML e Ingegneria del software  
Ed. Mokabyte 2002  
<http://www.mokabyte.it/umlbook/download.htm>

Ing. SW: il ruolo di UML - 282

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Bibliografia: UML for business

---

H. Eriksson et M. Penker  
Business Modeling with UML  
Ed. John Wiley & Sons., 2000

A.G. Nillson et al.  
Perspective on Business Modeling  
Ed. Springer-Verlag, 1999

## Bibliografia Web

---

- OMG
- <http://www.omg.org>
  
- Astrakan
- <http://www.astrakan.com>
  
- Open Training
- <http://www.opentraining.com>

## Bibliografia Web

---

Documenti reperibili in rete presso

- <http://www.uml.org>
- <http://www.mokabyte.it/umlbook/index.htm>
- <http://www.ebxml.com>
- <http://www.oasis-open.org>
- <http://www.ibm.com>
- <http://www.microsoft.com>
- <http://www.iona.com>