

# Introduzione a UML

versione 16 marzo 2009

©Adriano Comai

<http://www.analisi-disegno.com>

# Obiettivo di questa introduzione

- fornire alcuni elementi di base su UML
- introdurre i diagrammi
- fornire indicazioni sulle modalità di utilizzo di UML

⇒ questi temi sono approfonditi, con esercitazioni, nel corso “Sviluppo di applicazioni con UML”:

[http://www.analisi-disegno.com/a\\_comai/corsi/sk\\_uml.htm](http://www.analisi-disegno.com/a_comai/corsi/sk_uml.htm)

# Unified Modeling Language (UML)

- linguaggio (e notazione) universale, per rappresentare qualunque tipo di sistema (software, hardware, organizzativo, ...)
- standard OMG (Object Management Group), dal nov.1997
- originatori:
  - Grady Booch
  - Ivar Jacobson
  - Jim Rumbaugh

# Cos'è UML (e cosa non è)

- è un linguaggio di rappresentazione dei sistemi
- serve a specificare le caratteristiche di un nuovo sistema, oppure a documentarne uno già esistente
- è uno strumento di comunicazione tra i diversi ruoli coinvolti nello sviluppo e nell'evoluzione dei sistemi

# UML non è una “metodologia”

- è un linguaggio, non un metodo completo
- notazione, sintassi e semantica sono standard
- ma UML non è legato ad uno specifico processo, e non fornisce indicazioni sul proprio utilizzo
- quindi può essere (ed è) utilizzato da persone e gruppi che seguono approcci diversi (è “indipendente dai metodi”)

# UML come standard

- è stato definito con il contributo di molti metodologi, e delle più importanti società di software mondiali
- la sua evoluzione è a carico dell'OMG, e soggetta a procedure ben definite per ogni cambiamento
  - versione attuale: **2.2**
  - documenti ufficiali: <http://www.omg.org>

# Storia versioni di UML

novembre 1997:	1.1
dicembre 1998:	1.2
giugno 1999:	1.3
maggio 2001:	1.4
marzo 2003:	1.5
agosto 2005 :	2.0
aprile 2006:	2.1
maggio 2008:	2.2

# Diagrammi UML (versioni 1.x)

diagrammi “strutturali”:

diagramma delle classi (class)

diagramma dei componenti (component)

diagramma di distribuzione (deployment)

} implementation

diagrammi “comportamentali” :

diagramma dei casi d’uso (use case)

diagramma di sequenza (sequence)

diagramma di collaborazione (collaboration)

diagramma di stato (statechart)

diagramma delle attività (activity)

} interaction

# Diagrammi UML (versioni 2.x)

## diagrammi “strutturali”:

- diagramma delle classi (class)
- diagramma degli oggetti (object)
- diagramma dei componenti (component)
- diagramma delle strutture composite (composite structure)
- diagramma di deployment (deployment)
- diagramma dei package (package)

hanno molti  
aspetti in comune

## diagrammi “comportamentali”:

- diagramma dei casi d’uso (use case)
- diagramma di stato (statechart)
- diagramma delle attività (activity)

## diagrammi “comportamentali di interazione”:

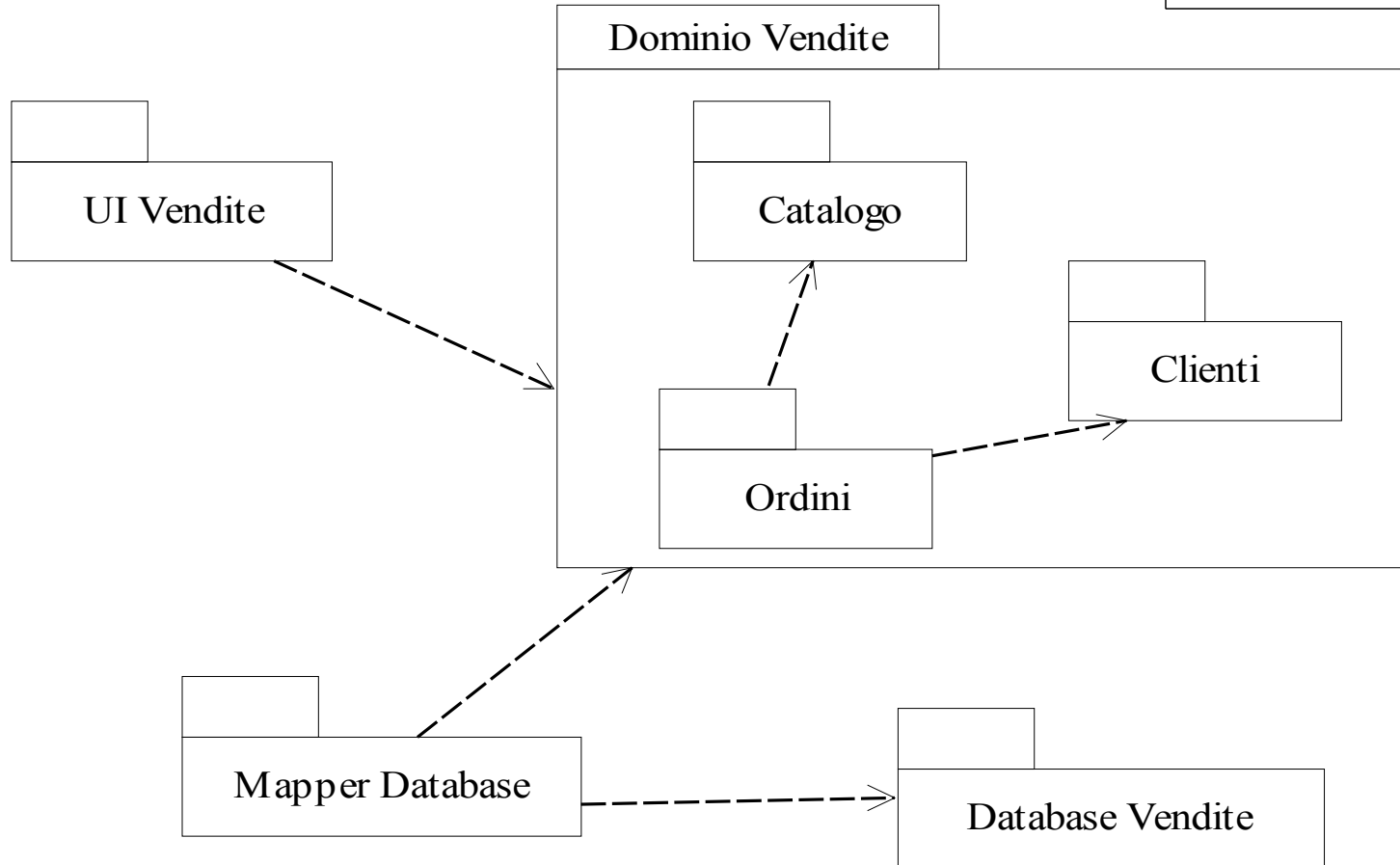
- diagramma di sequenza (sequence)
- diagramma di comunicazione (communication)
- diagramma dei tempi (timing)
- diagramma di sintesi dell’interazione (interaction overview)

hanno molti  
aspetti in comune

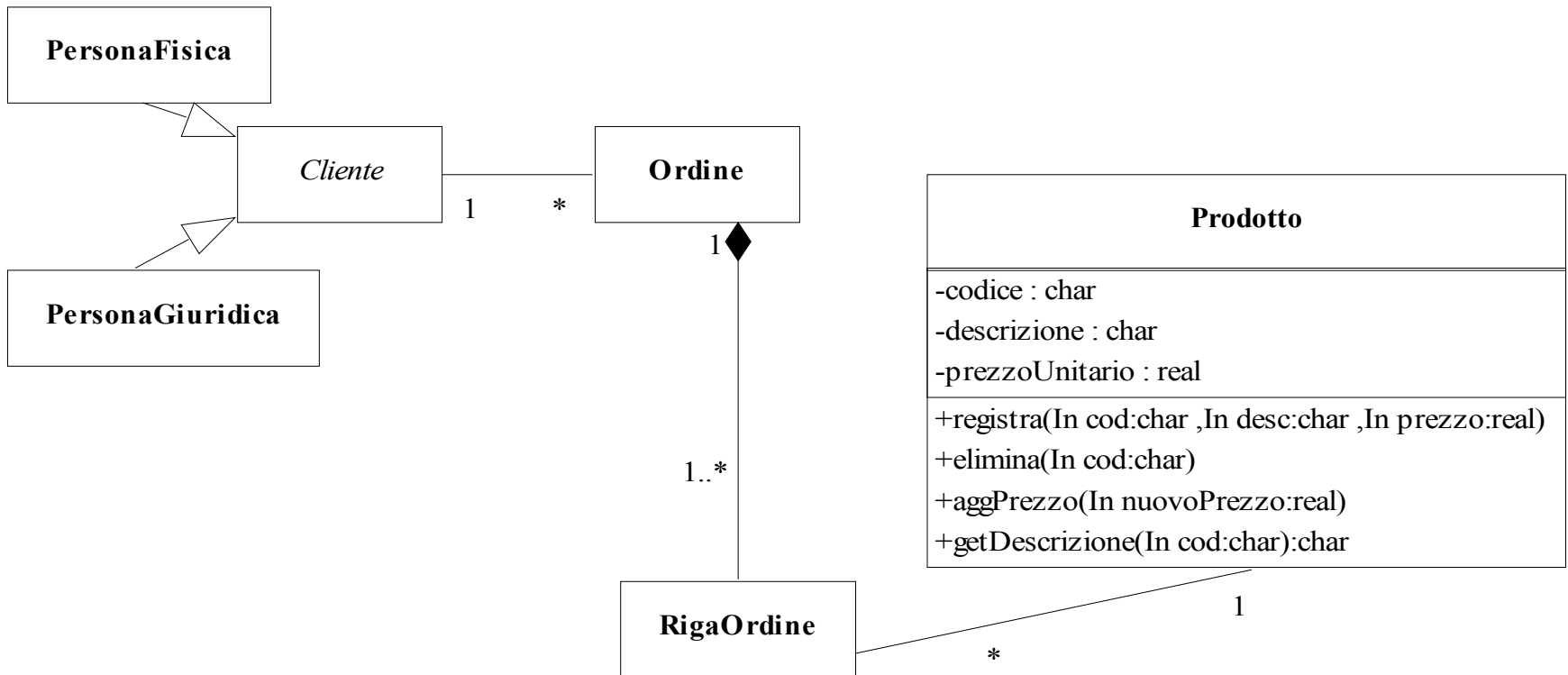
# Package diagram

UML 2.0

in UML 1.x non è un diagramma "ufficiale"



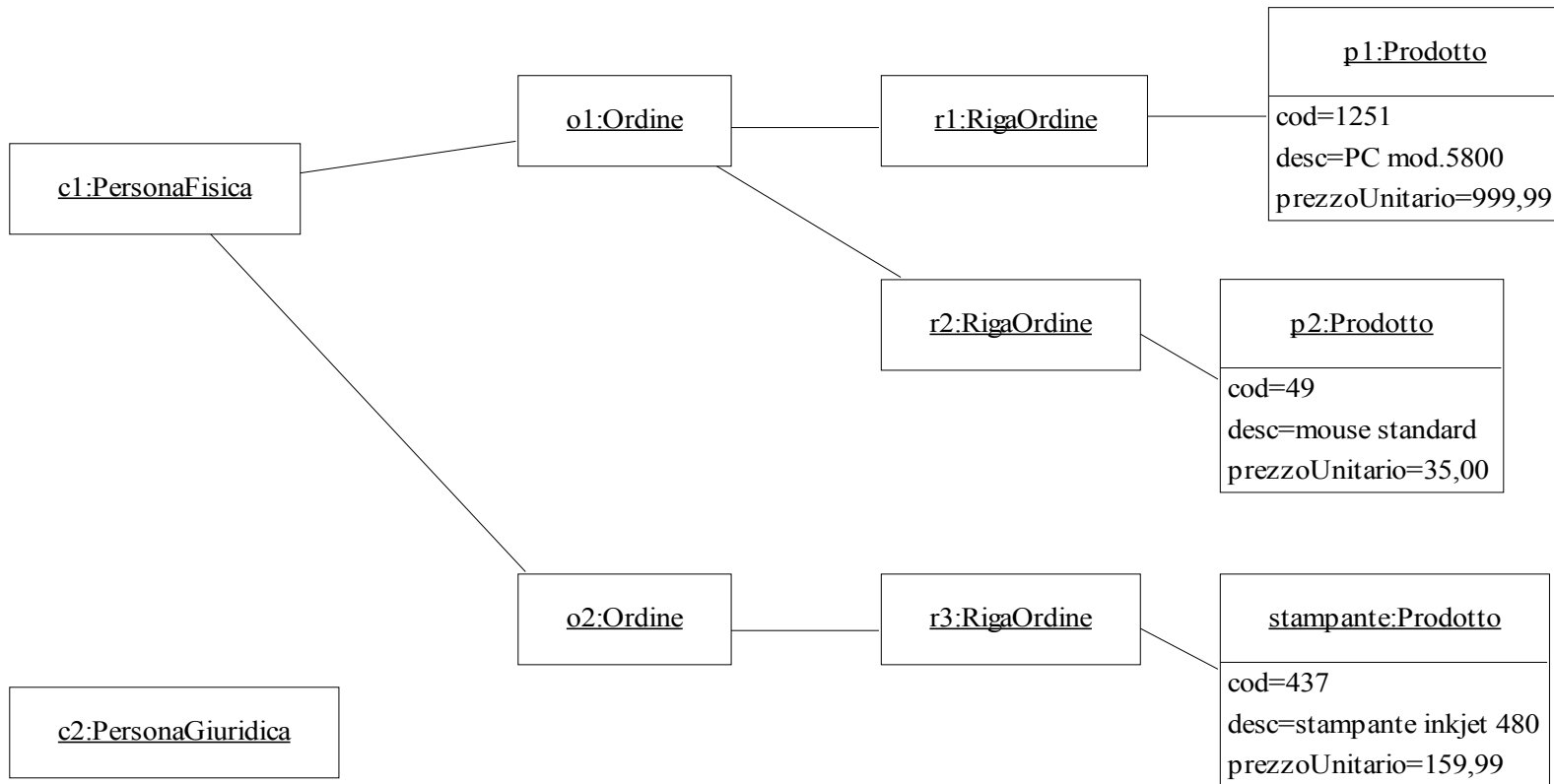
# Class diagram



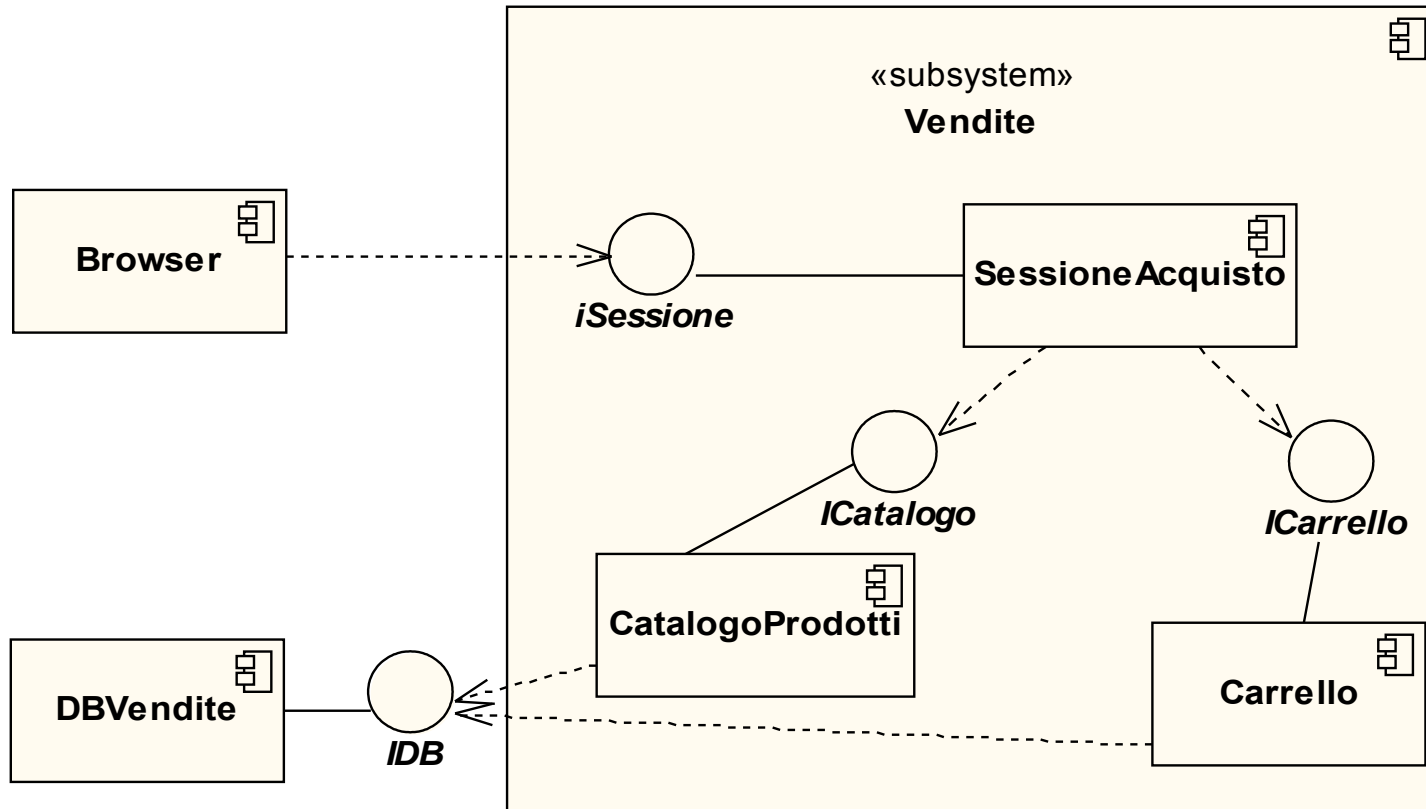
# Object diagram

UML 2.0

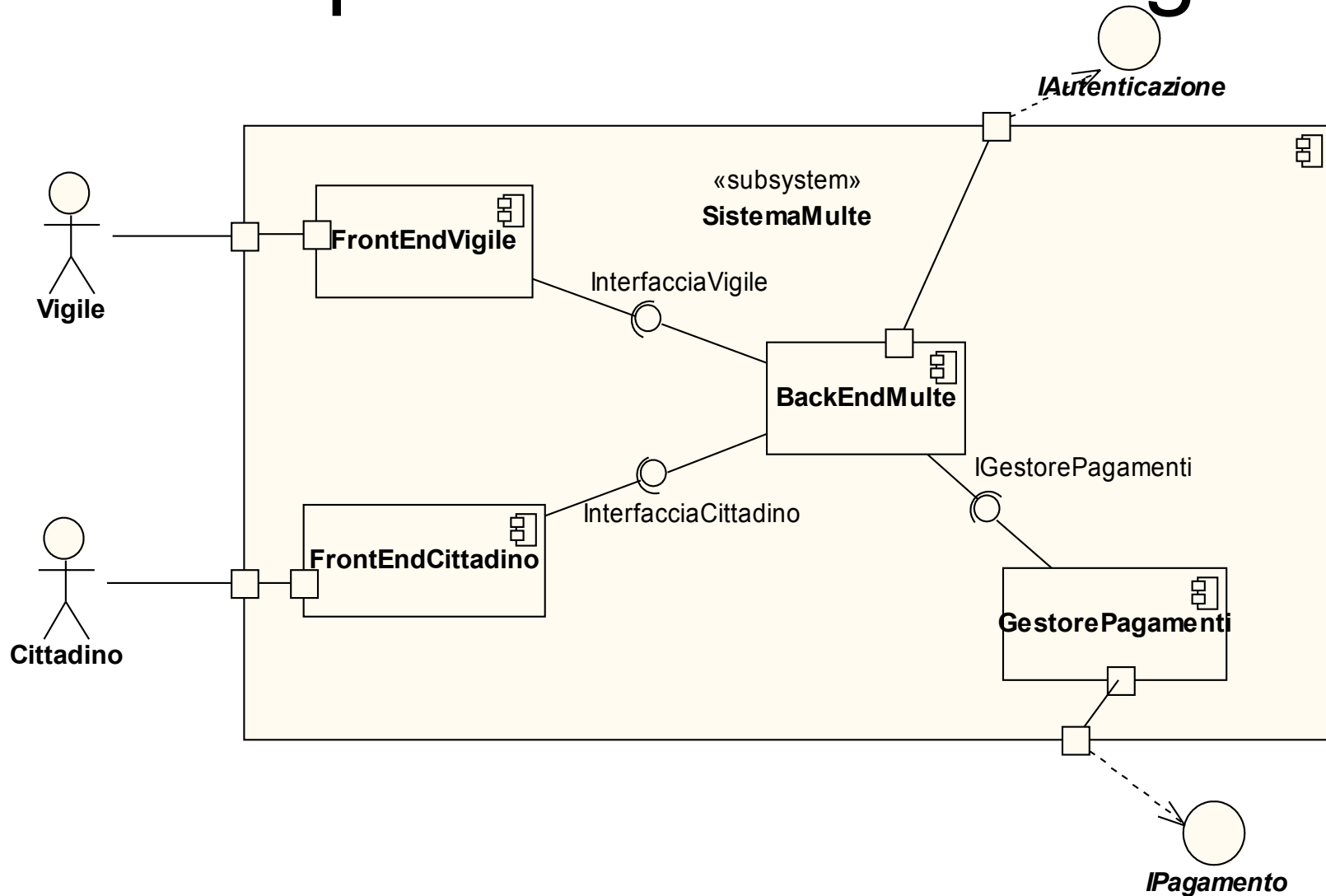
in UML 1.x non è un diagramma "ufficiale"



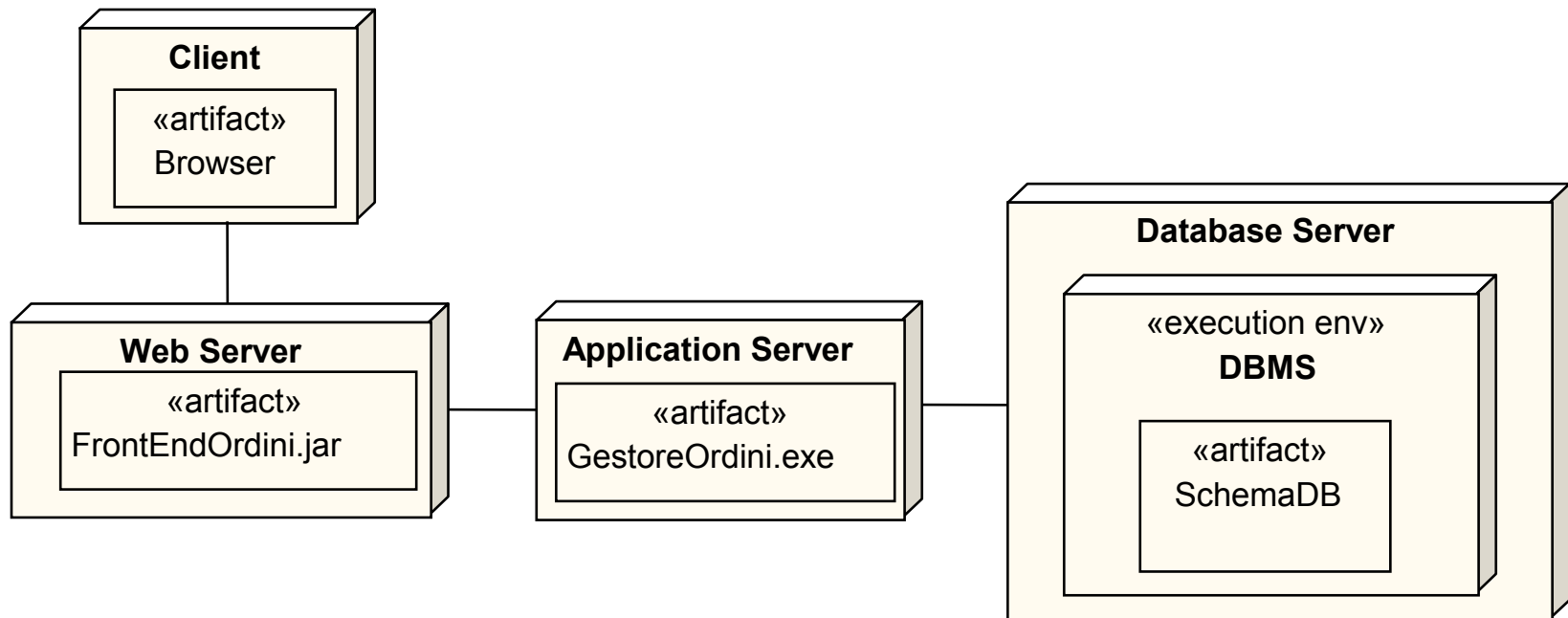
# Component diagram



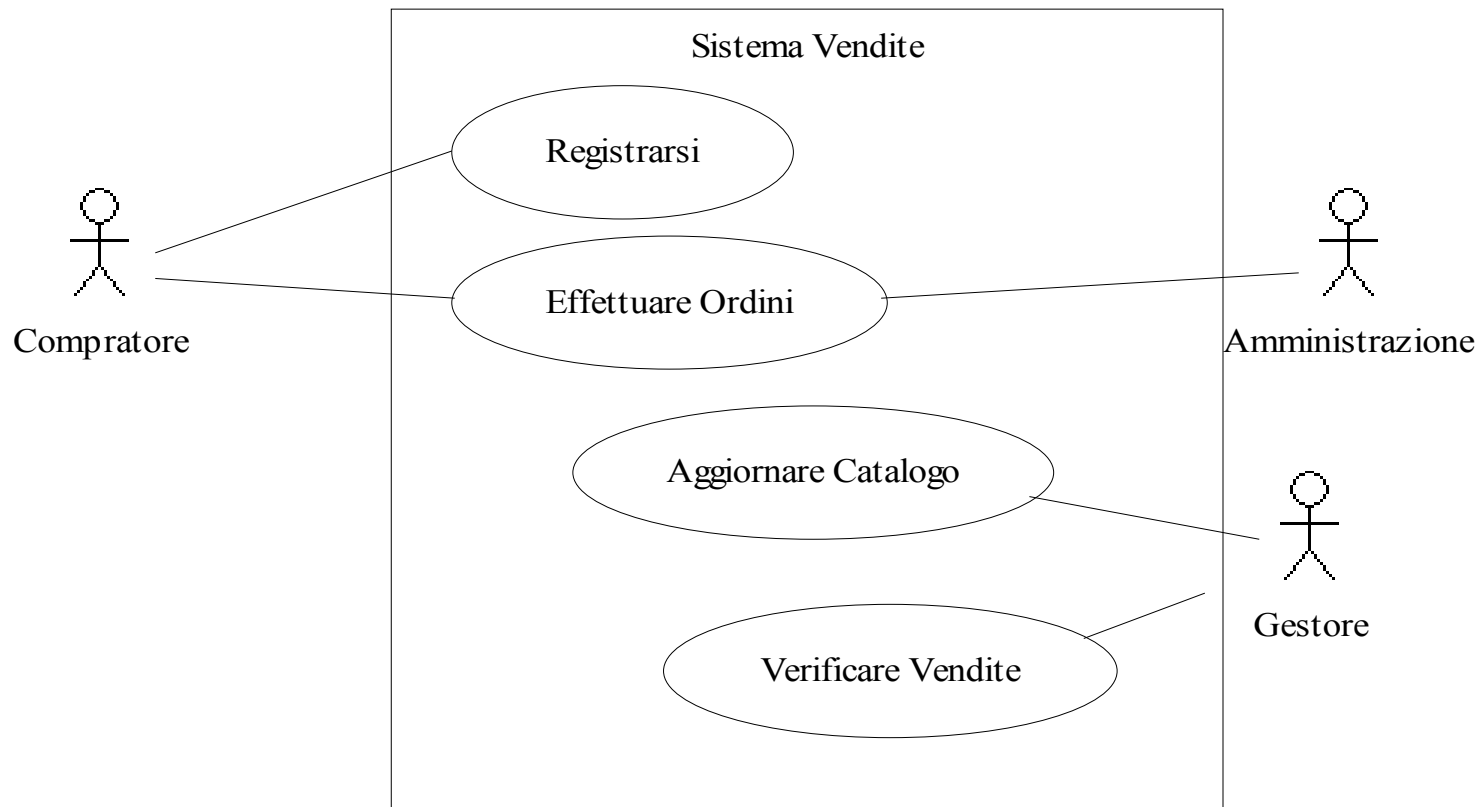
# Composite structure diagram



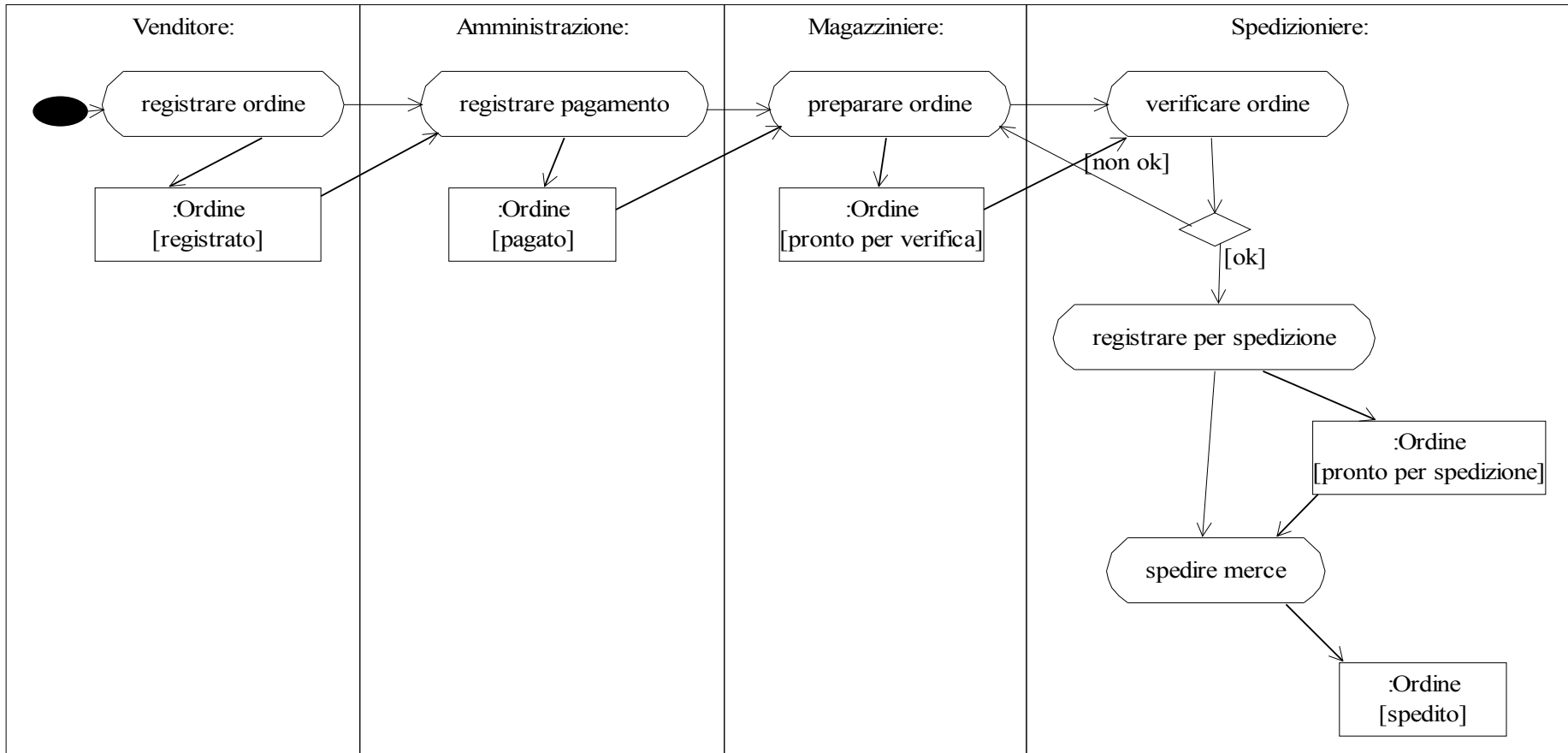
# Deployment diagram



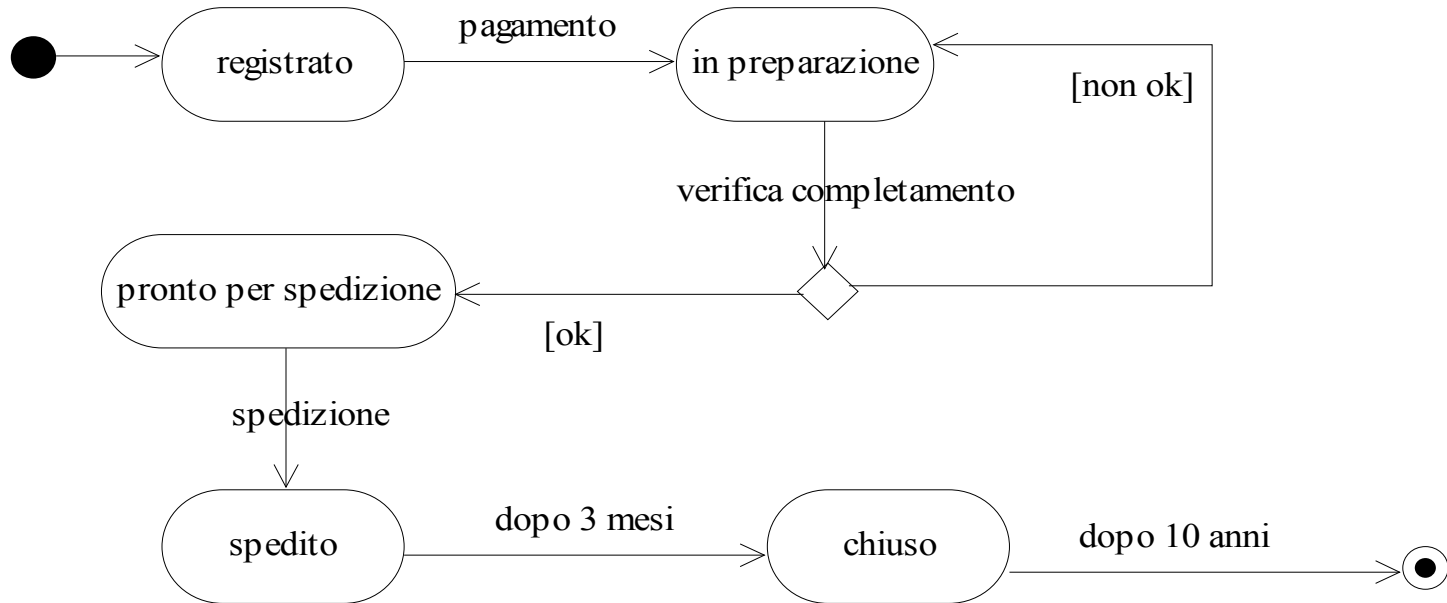
# Use case diagram

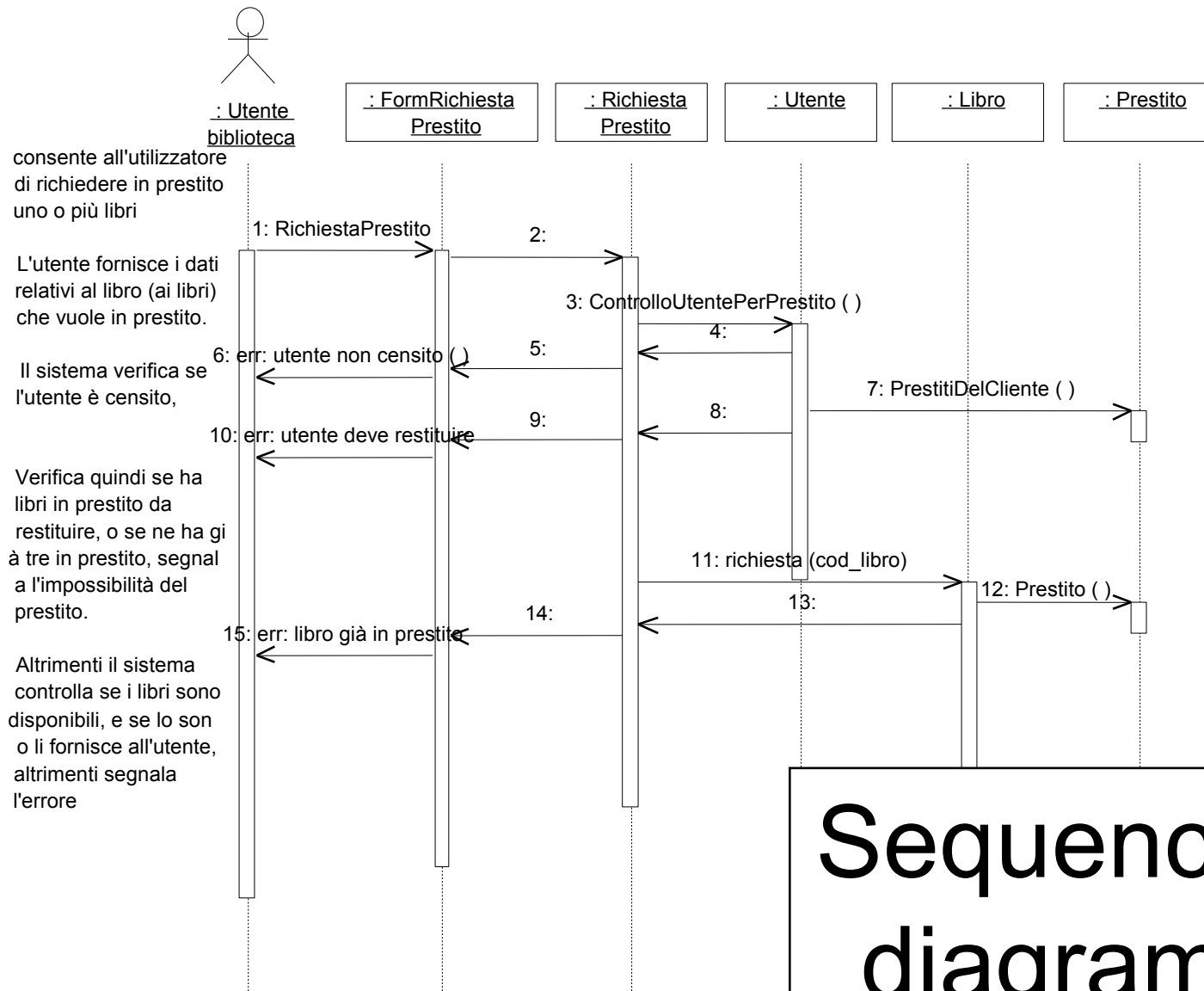


# Activity diagram



# Statechart diagram

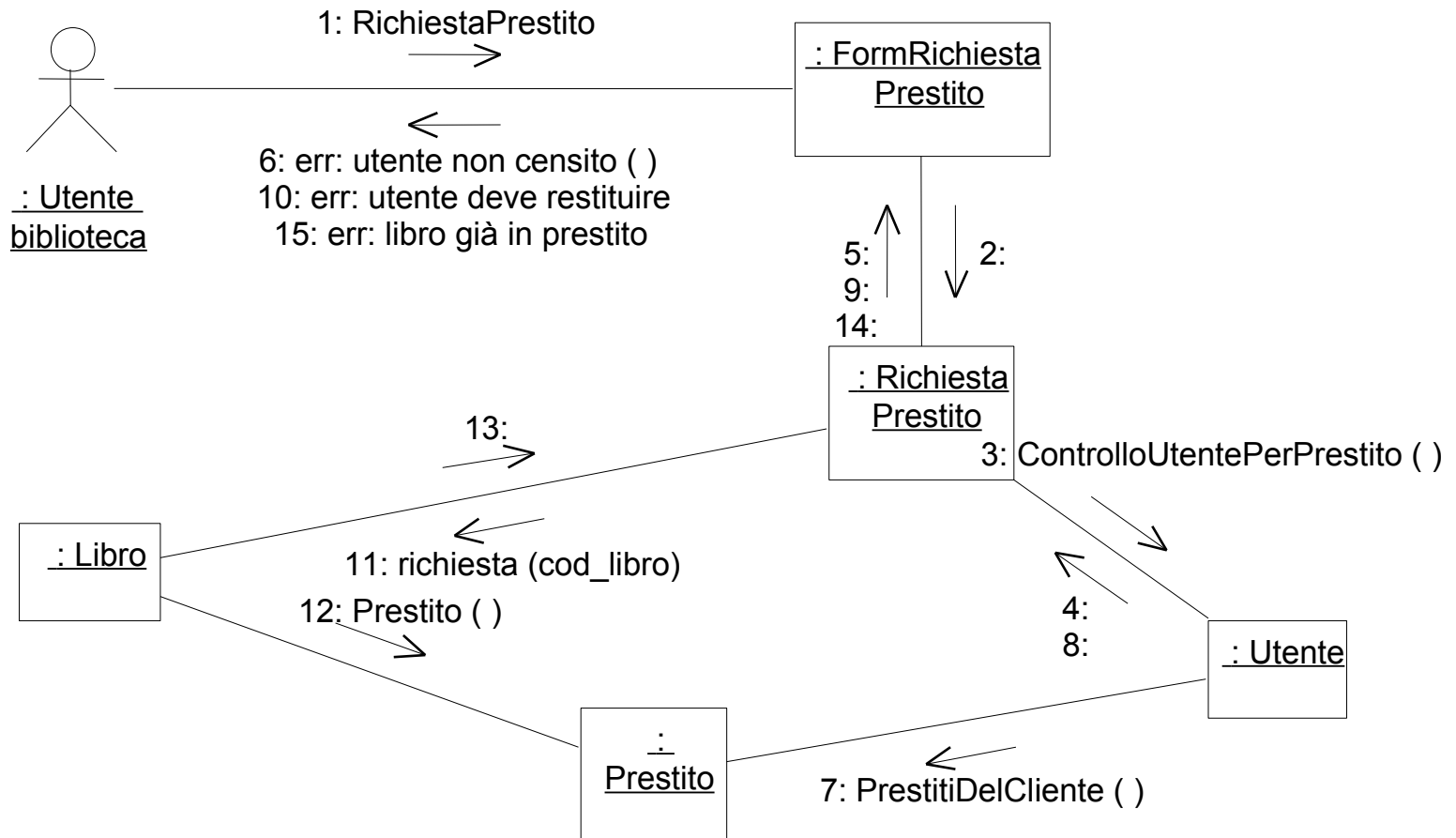




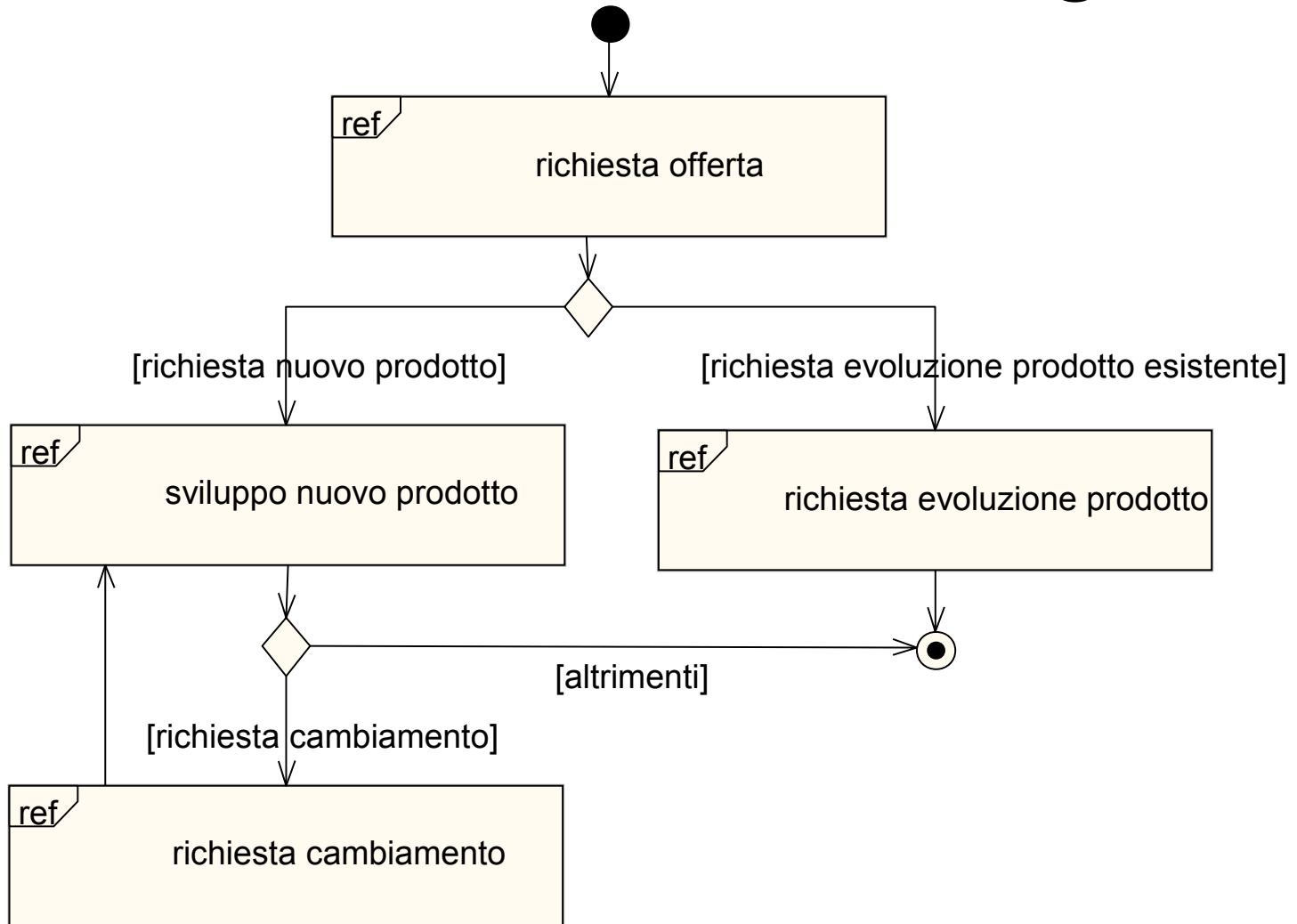
# Sequence diagram

in UML 1.x si chiama "collaboration"

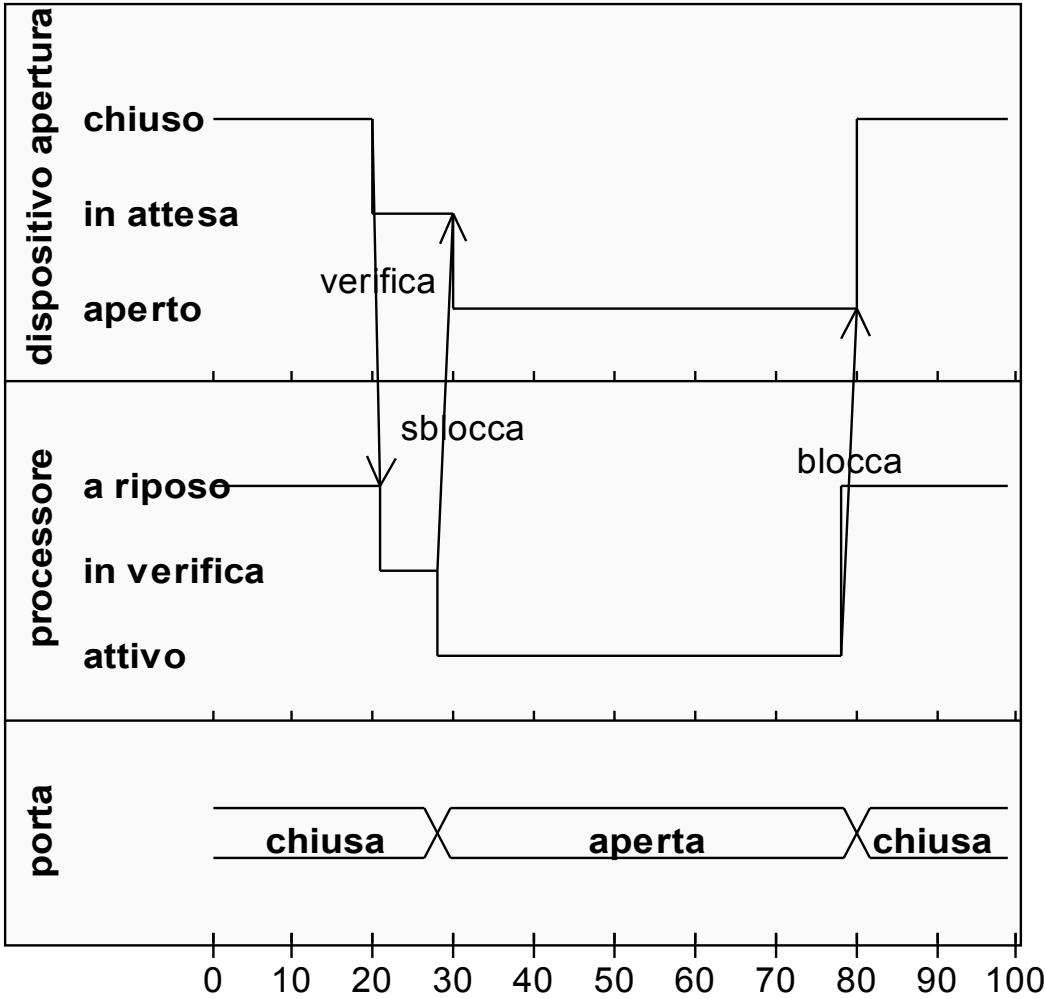
# Communication diagram



# Interaction overview diagram



# Timing diagram



# UML: meta-modello e diagrammi

- UML è basato su un meta-modello integrato, composto da numerosi elementi, collegati tra loro secondo regole precise
- utilizzando gli elementi del meta-modello è possibile creare i modelli per i sistemi da rappresentare
- molti elementi hanno una icona che li rappresenta graficamente
- gli elementi del meta-modello possono comparire in diagrammi di diverso tipo
- le regole permettono verifiche di correttezza

# Strumenti: visual modeling UML

Una lista molto parziale (ne esistono decine...)

- Rose; Rational Modeler ed Architect (IBM - Rational)
- Together (Borland)
- Visio (Microsoft)
- TAU (IBM - Telelogic)
- Objecteering (Softeam)
- Poseidon (Gentleware)
- Enterprise Architect (Sparx Systems)
- Magic Draw (No Magic)
- Argo (open source)
- StarUML (open source)

# Strumenti UML: criteri di scelta

Tra i fattori da considerare:

- costo
- aderenza allo standard UML
- supporto al lavoro di gruppo
- generazione codice / reverse engineering
- integrazione con altre tipologie di strumenti:
  - gestione requisiti
  - programmazione (IDE)
  - gestione test
  - gestione configurazione

# UML è complesso

- intende rappresentare qualunque tipo di sistema software, a diversi livelli di astrazione
- il numero degli elementi è elevato, e in molti casi è possibile scegliere tra forme di rappresentazione diverse
- UML non suggerisce, né tantomeno prescrive una sequenza di realizzazione dei diversi diagrammi
- offre un'ampia gamma di possibili modalità di utilizzo, tra le quali i progettisti sono liberi di scegliere

# UML va adattato alle proprie esigenze

Tra i fattori da considerare:

- settore di attività (es. militare, finanziario)
- tipologia di progetto (rischio, complessità)
- processo di sviluppo adottato
- esigenze di conformità a norme e standard
- comunicazione con committenti e stakeholders
- comunicazione con fornitori
- composizione e distribuzione del gruppo di lavoro

⇒ non ha senso che tutti usino UML nello stesso modo

# UML in sintesi

- è uno standard: uniformità nei concetti e nelle notazioni utilizzate, interoperabilità tra strumenti di sviluppo, indipendenza dai produttori, dalle tecnologie, dai metodi
- è articolato: può rappresentare qualunque sistema software, a diversi livelli di astrazione
- è complesso: va adattato ("ritagliato") in base alle specifiche esigenze dei progettisti e dei progetti, utilizzando **solo ciò che serve nello specifico contesto**

# Bibliografia

- Jim Rumbaugh, Ivar Jacobson, Grady Booch: ***The Unified Modeling Language Reference Manual*** (2nd Edition) - Addison Wesley 2005
- Bruce Powel Douglass : ***Real Time UML*** (3rd Edition) - Addison Wesley 2004
- Martin Fowler : ***UML Distilled*** (3rd Edition) - Addison Wesley 2003
- Robert Martin : ***UML for Java Programmers*** - Prentice Hall 2003
- Craig Larman : ***Applying UML and Patterns*** (3rd Edition) - Prentice Hall 2005

⇒ <http://www.omg.org>

Per approfondimenti e altri materiali:

<http://www.analisi-disegno.com>