Unified Modeling Language (3)

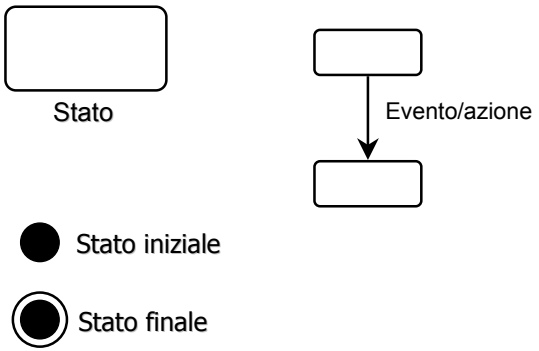
A cura di Luciano Baresi

Statechart diagram

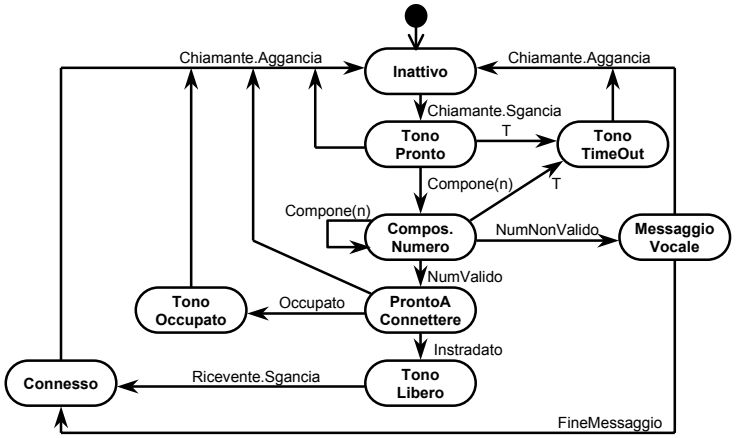
State Diagram

- Rappresentano il comportamento dei singoli oggetti di una classe in termini di
  - Eventi a cui gli oggetti (la classe) sono sensibili
  - Azioni prodotte
  - Transizioni di stato
    - Identificazione degli stati interni degli oggetti
- Possibilità di descrivere evoluzioni parallele
- Sintassi mutuata da StateChart (D. Harel)

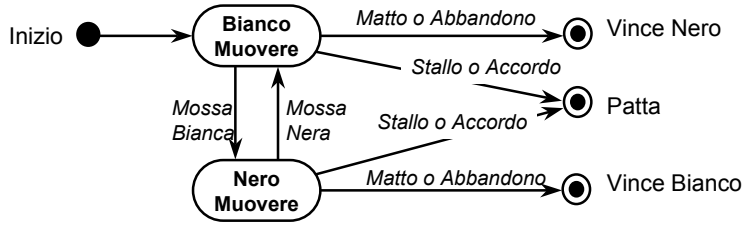
Elementi grafici



Esempio
(Telefonata)

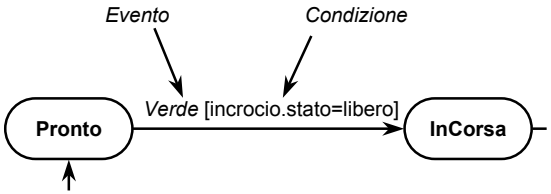


Inizio e Fine



Condizioni

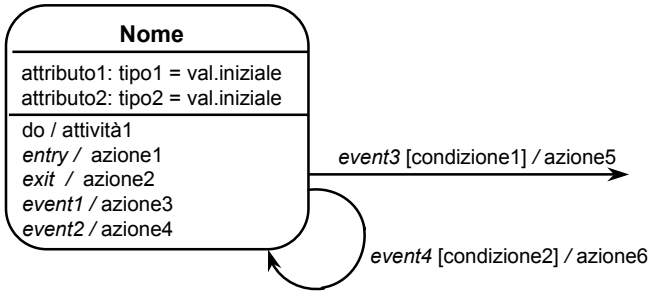
- Funzioni booleane sui valori degli oggetti
- Utili quando non basta l'evento, ma si vuole aggiungere un predicato



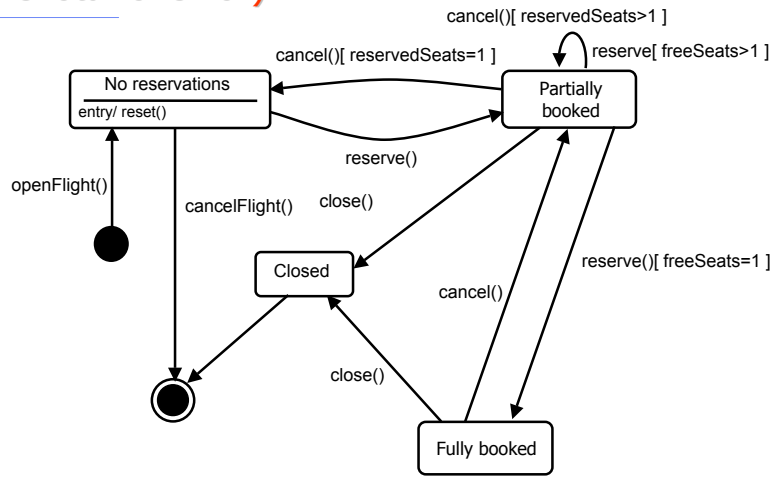
Operazioni

- Azioni
  - Operazioni che hanno durata istantanea
    - Tipicamente produzione di eventi
  - Sono associate alle transizioni di stato oppure all'ingresso o all'uscita da uno stato
- Attività
  - Sono operazioni con durata significativa
  - Sono associate ad uno stato
    - Continue o sequenziali

Stato Completo

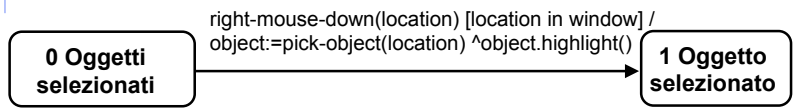


Esempio
(Prenotazione voli)

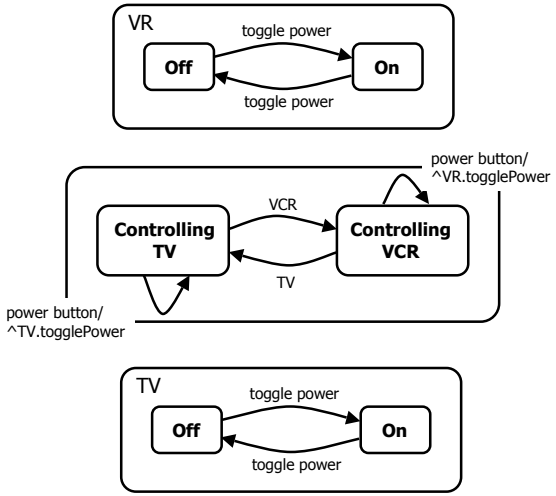


Eventi
(Generati da Azioni)

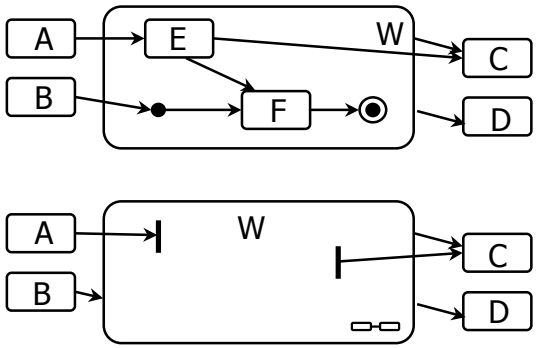
- Spesso le azioni consistono nell'inviare un evento ad un altro oggetto
  - evento(arg) [cond] / azione ^target.azione(arg)



Esempio
(TV, Video Registratore e Telecomando)

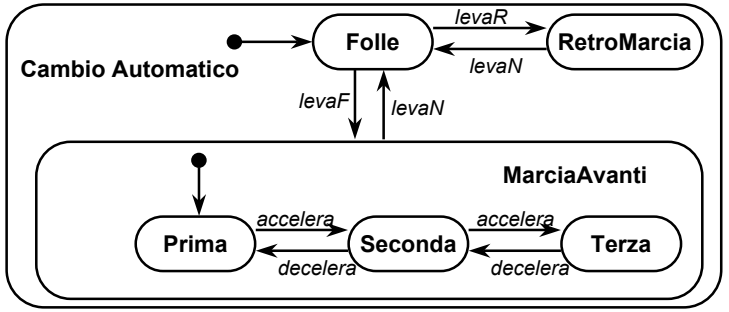


Stub
(Sottostati nascosti)

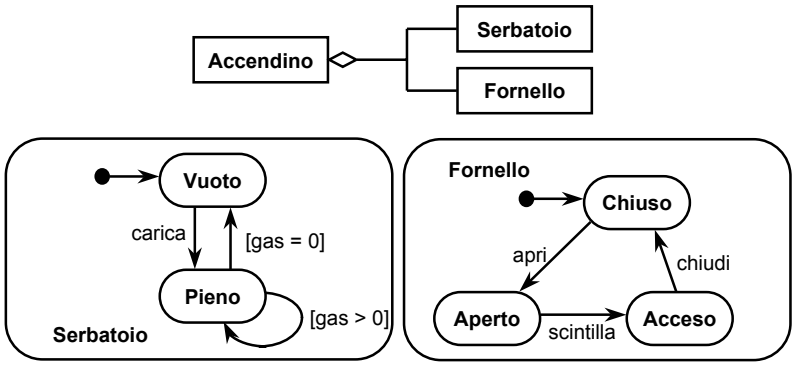


Decomposizione OR

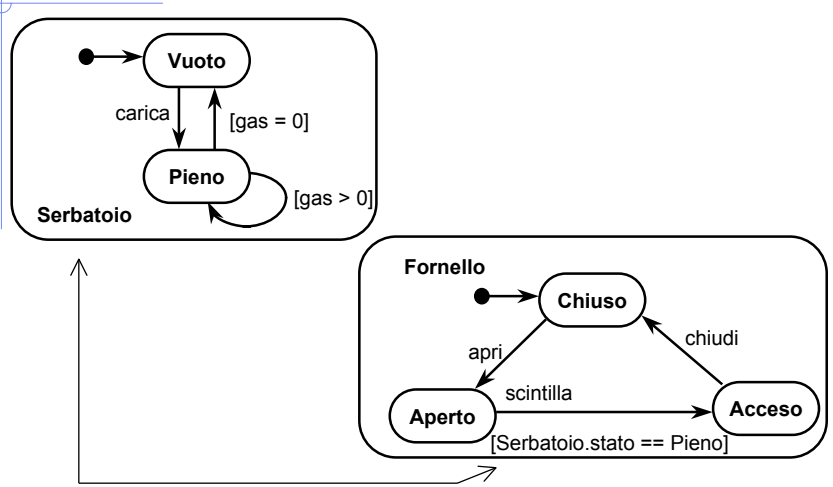
- Un macro stato equivale ad una scomposizione OR degli stati
- I sottostati ereditano le transizioni dei loro superstati



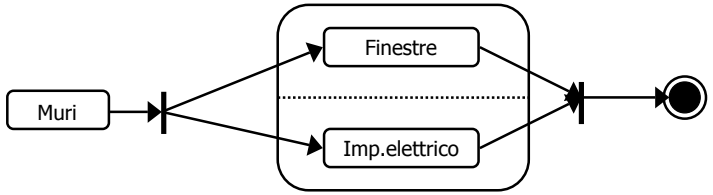
Oggetti Compositi



Interazione fra Componenti

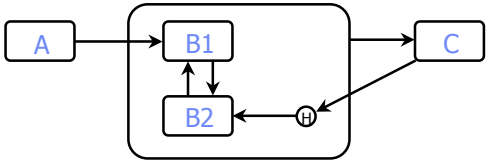


Decomposizione AND

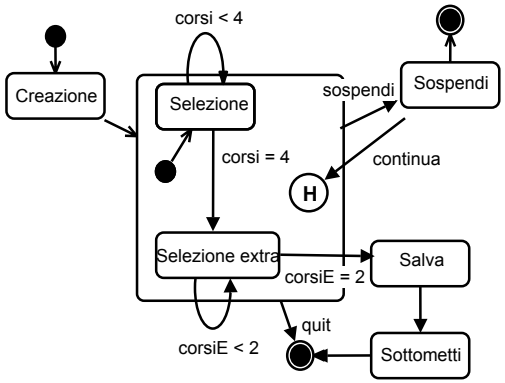


History

- Quando l'esecuzione lascia uno stato S con history
  - Si salva l'ultimo stato visitato S_1 in S
- Quando l'esecuzione ritorna in S
  - Si riparte da S_1



Example
(Selezione corsi)



Activity Diagram

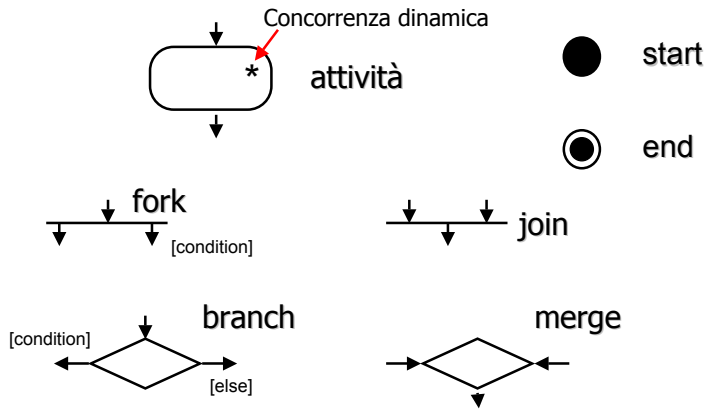
Activity Diagram

- Forniscono la sequenza di operazioni che definiscono un'attività più complessa
- Possono essere considerati State Diagram particolari
  - Ogni stato contiene (è) un'azione
- Un Activity Diagram può essere associato
  - A una classe
  - All'implementazione di un'operazione
  - Ad uno Use Case

Activity Diagram (cont.)

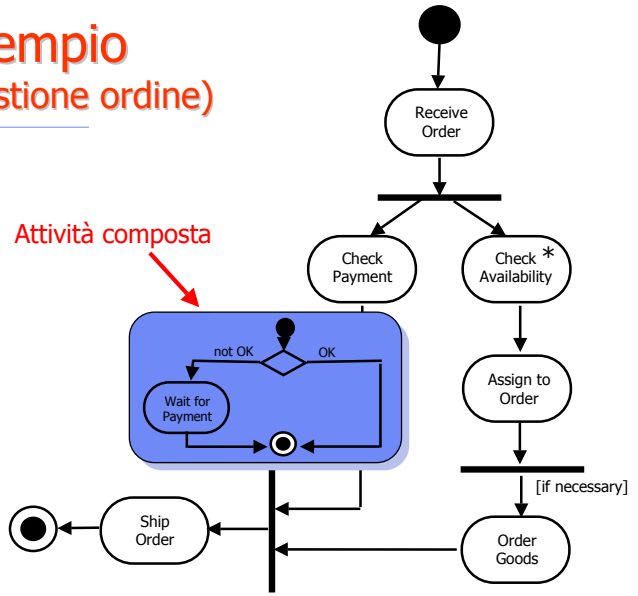
- Derivano da event diagrams e reti di Petri
- Utili per modellare
  - comportamenti sequenziali
  - non determinismo
  - concorrenza
  - sistemi distribuiti
  - business workflow
  - operazioni

Elementi grafici



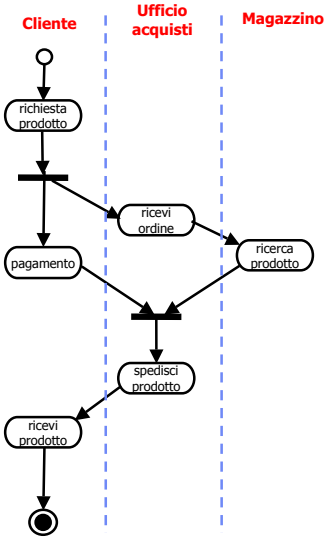
Le attività possono essere gerarchiche

Esempio
(Gestione ordine)

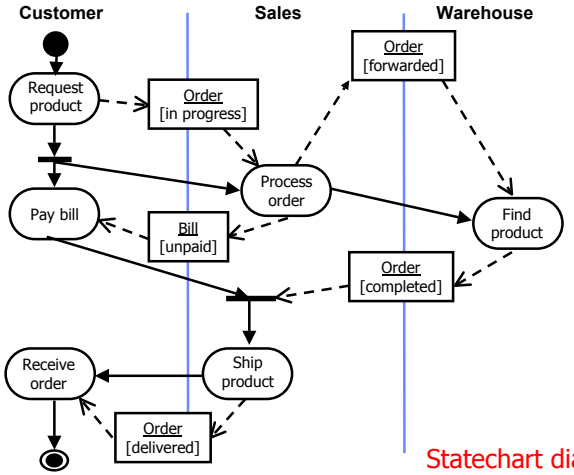


Swimlane
(Corsie)

- Identificano le responsabilità relative alle diverse operazioni
  - Parti di un oggetto
  - Oggetti diversi
- In un Business Model identificano le unità organizzative



Attività e flussi degli oggetti



Statechart diagram

Implementation Diagram

Implementation Diagram

- Component Diagram
  - Definiscono i componenti "fisici" che realizzano l'applicazione (eseguibili, librerie, ecc..)
- Deployment Diagram
  - Definiscono il partizionamento (la struttura) dell'applicazione: processi e/o processori

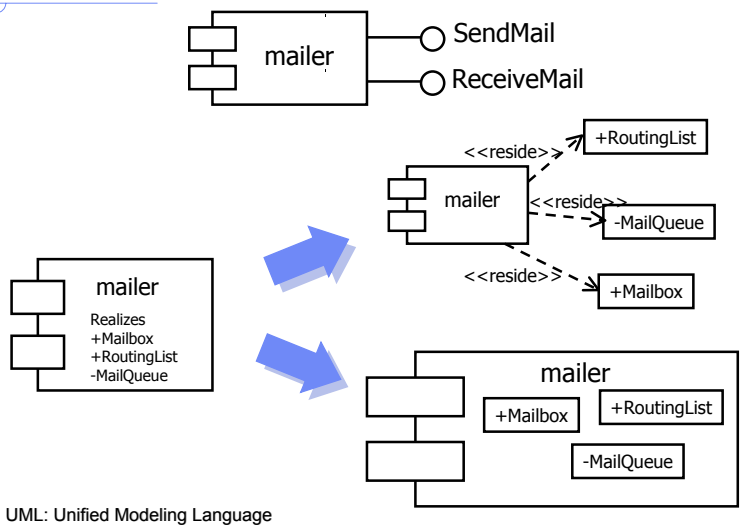
Component diagram

- Definiscono le relazioni fra i componenti software che realizzano l'applicazione
  - sorgenti, binari, eseguibili
- I diversi componenti offrono e usano interfacce specifiche
- Primo passo verso *Component Programming*

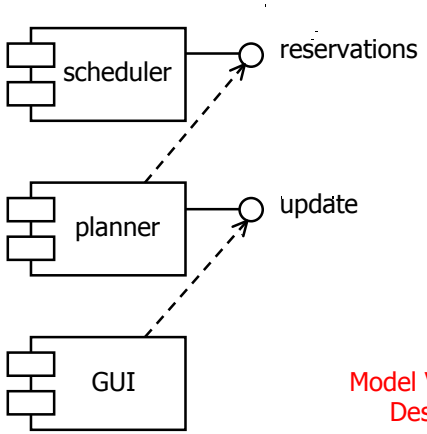
Component diagram (cont.)

- Rappresentano l'implementazione del sistema
- Parte della specifica architetturale
- Serve per
  - Organizzare il codice
  - Vincolare (definire) una versione particolare
  - Definire una base di dati

Componenti

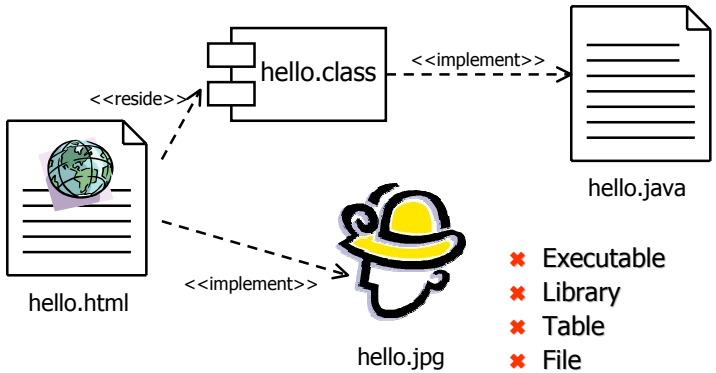


Esempio
(Scheduler)



Model View Controller
Design pattern

Usando icone carine
(HelloWorld)



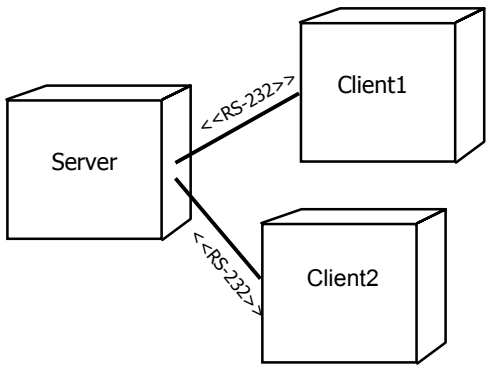
- Executable
- Library
- Table
- File
- Document
- ...

Deployment Diagram

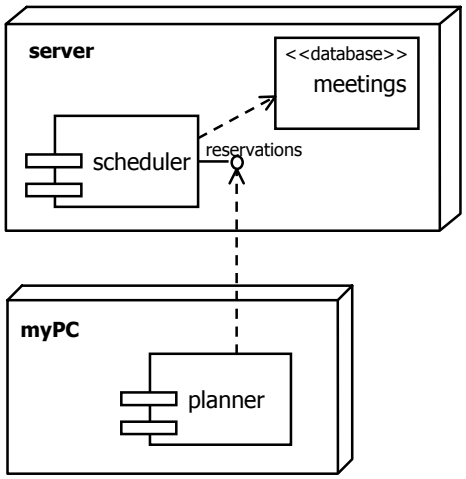
- Definiscono la configurazione a run-time dei componenti identificati al passo precedente
- Identificano processi e processori (nodi) dell'applicazione

Associazioni

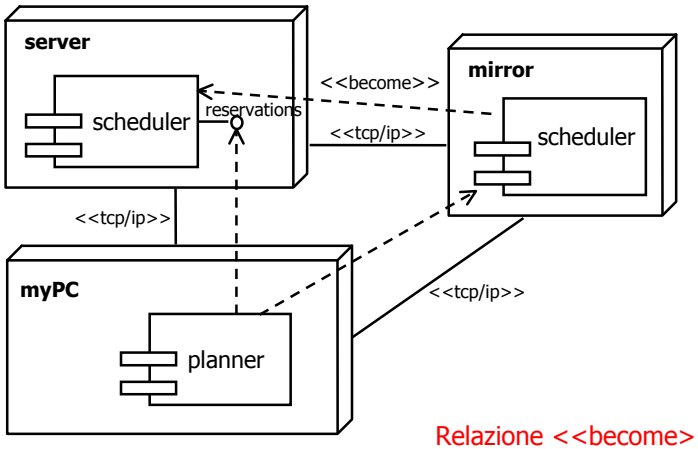
- Rappresentano le connessione fisiche tra i nodi



Nodi e componenti
(I soluzione)



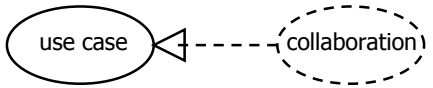
Nodi e componenti
(II soluzione)



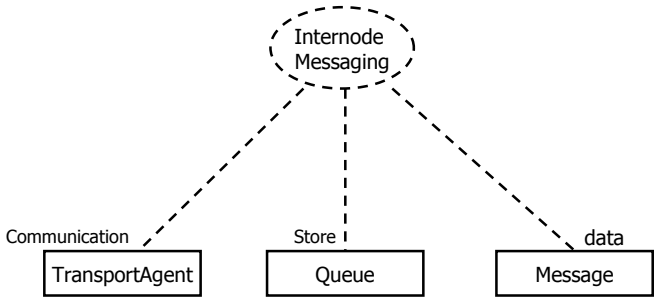
Relazione <<become>>

Il 10mo modello (Collaboration)

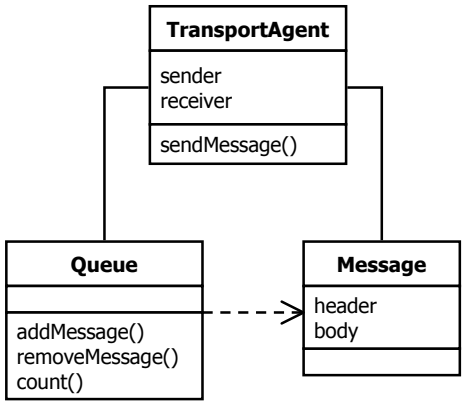
- Collaboration sono gruppi di classi, oggetti e altri elementi che *lavorano insieme* per risolvere un problema
  - Possiamo dire che i collaboration definiscono pattern
  - Comprendono
    - Una parte strutturale (statica) che definisce gli elementi della soluzione
    - Una parte di comportamento (dinamica) per definire le interazioni
- Si può dire che i collaboration implementano gli use case



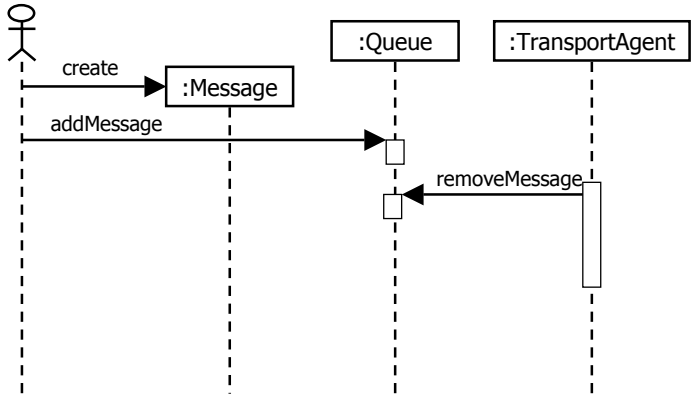
Collaboration



Collaboration-Static view



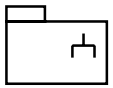
Collaboration-Dynamic view



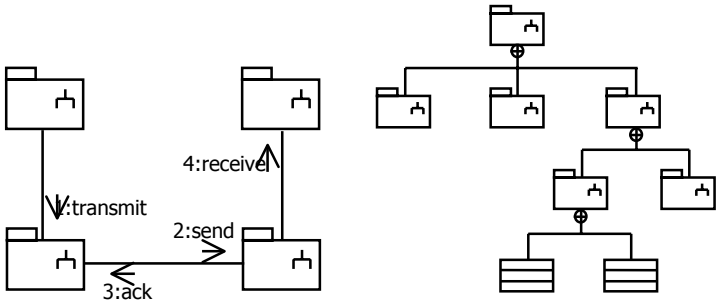
Gestione del modello

- Package
  - Servono per organizzare specifiche complesse
    - Visibilità, import, access
- Subsystem
  - Servono per strutturare sistemi complessi
    - Specifica e realizzazione
- Model
  - Servono per mostrare aspetti diversi
    - Tracciabilità

Subsystem



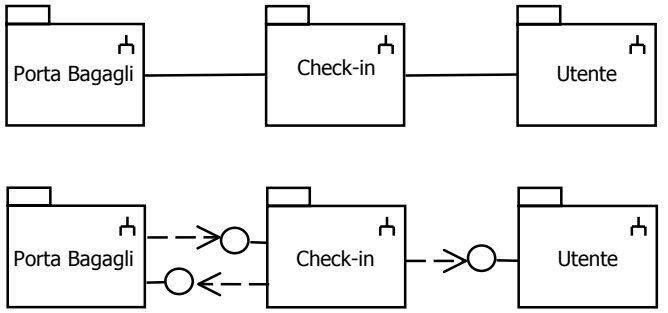
- Si usano per decomporre il sistema



Collaboration

Gerarchia

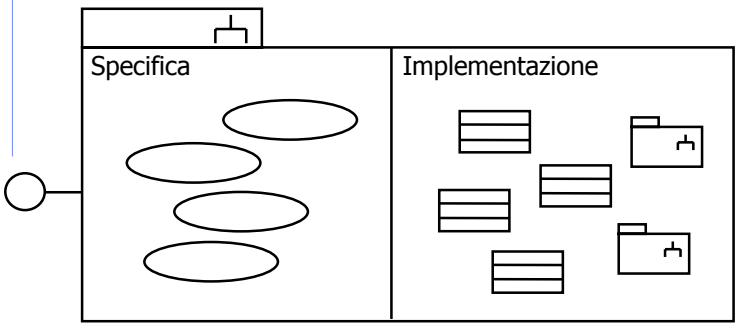
Interfacce



Due aspetti

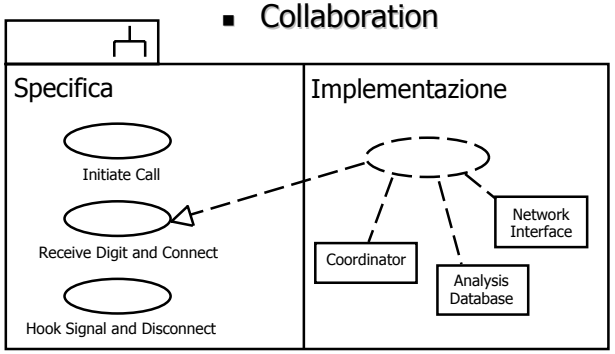
- Un sottosistema ha due aspetti
  - Visione esterna: servizi forniti all'esterno
    - Possibili approcci: use case, statecharts, classi logiche, …
  - Visione interna: come si realizzano i servizi
    - Possibili approcci: classi con le loro relazioni, gerarchia di sottosistemi e classi (come foglie), …
- C'è una corrispondenza tra i due aspetti

Specifica e implementazione

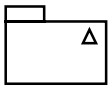


Corrispondenza

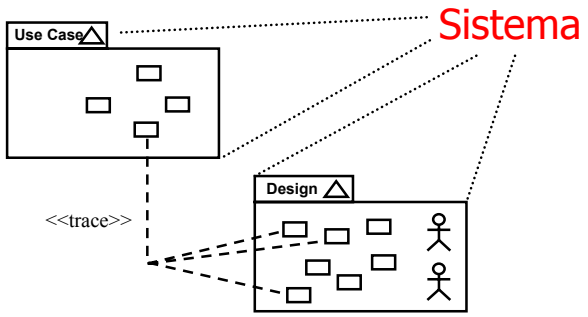
- Può essere espressa direttamente:
  - Relazioni di "implementazione"
  - Collaboration



Modello

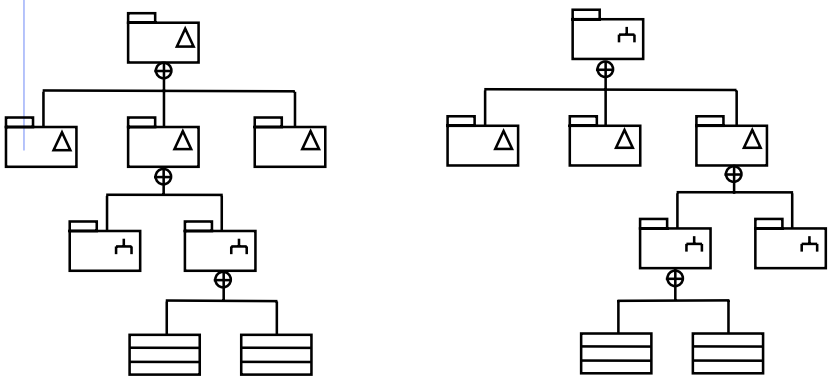


- Un modello è un'astrazione del sistema



Modelli e sottosistemi

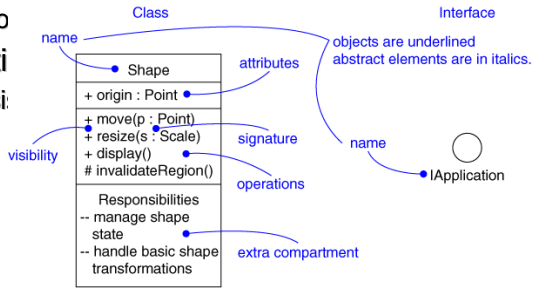
- Possono essere combinati in gerarchie



Riassumendo

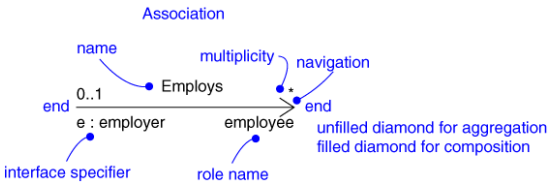
Elementi

- Elementi strutturali
  - classi, interfacce, collaboration, use case, classi attive, componenti, nodi
- Elementi "dinamici" (behavioral)
  - interaction, auto
- Raggruppamenti
  - package, sottosi
- Altri elementi
  - note



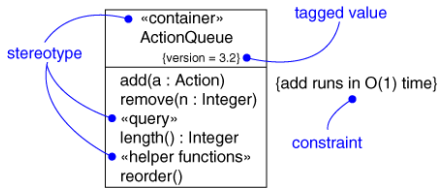
Relazioni

- Dipendenze
- Associazioni
- Generalizzazioni
- Realizzazioni



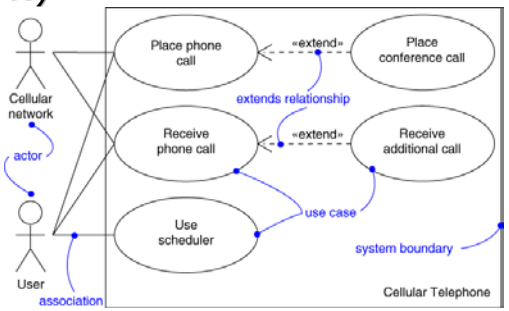
Estendibilità

- Stereotipi
- Restrizioni (tagged value)
- Vincoli (constraint)



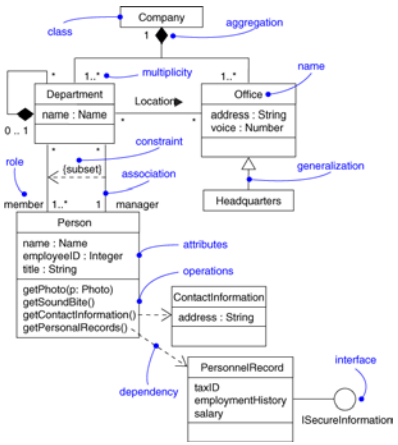
Use case diagram

- Catturano le funzionalità del sistema (vista utente)



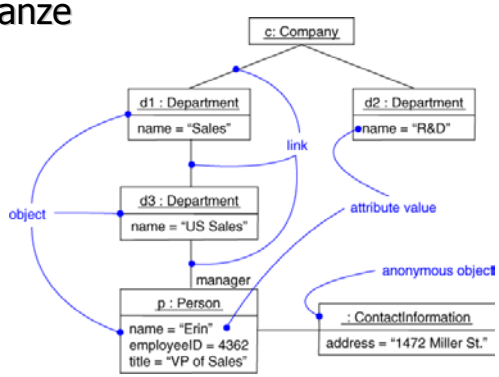
Class diagram

- Catturano il vocabolario del sistema



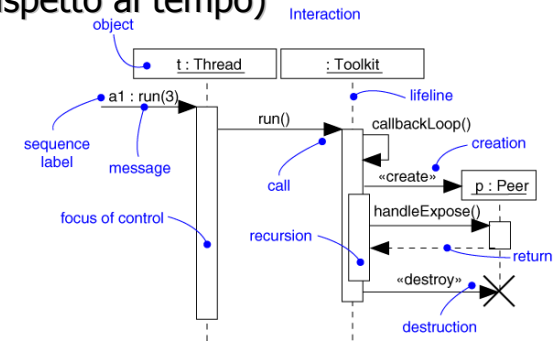
Object diagram

- Le istanze e i legami fra le istanze



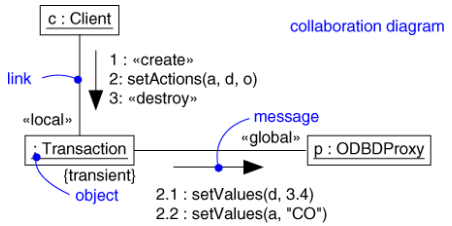
Sequence diagram

- Catturano il comportamento dinamico (rispetto al tempo)



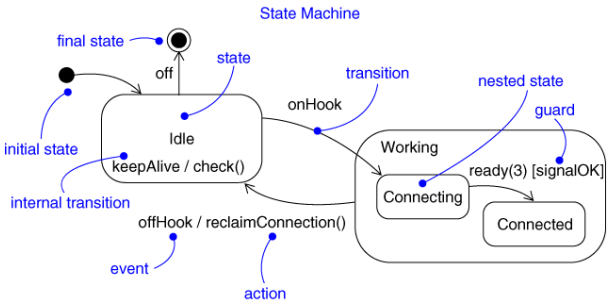
Collaboration diagram

- Catturano il comportamento dinamico (rispetto ai messaggi)



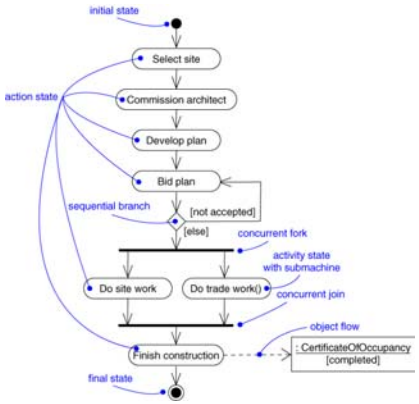
Statechart diagram

- Catturano il comportamento dinamico (rispetto agli eventi)



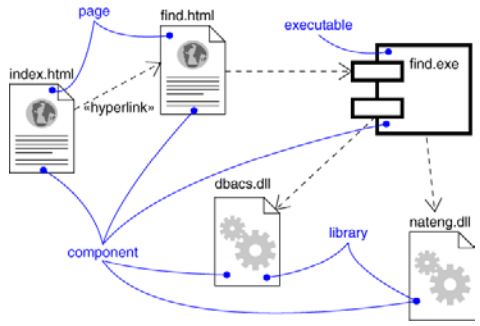
Activity diagram

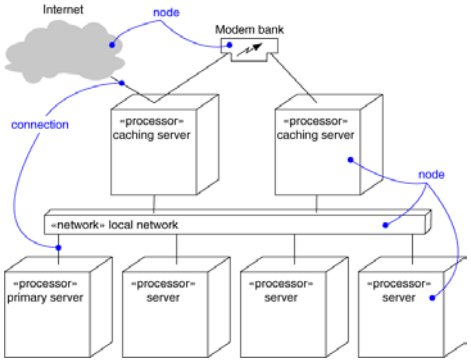
- Catturano il comportamento dinamico (rispetto attività)



Component diagram

- Catturano la struttura fisica dell'implementazione



Deployment diagram

- Catturano la topologia del sistema