

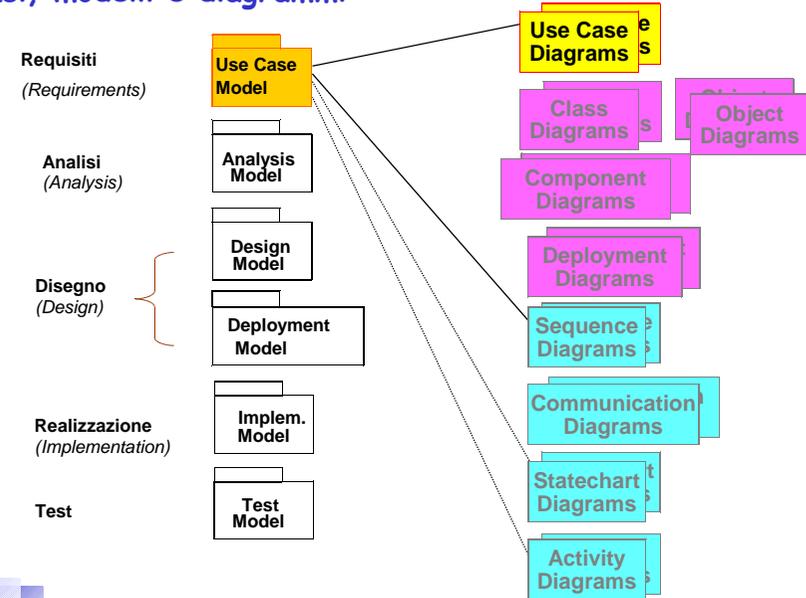
UML

Diagrammi casi d'uso

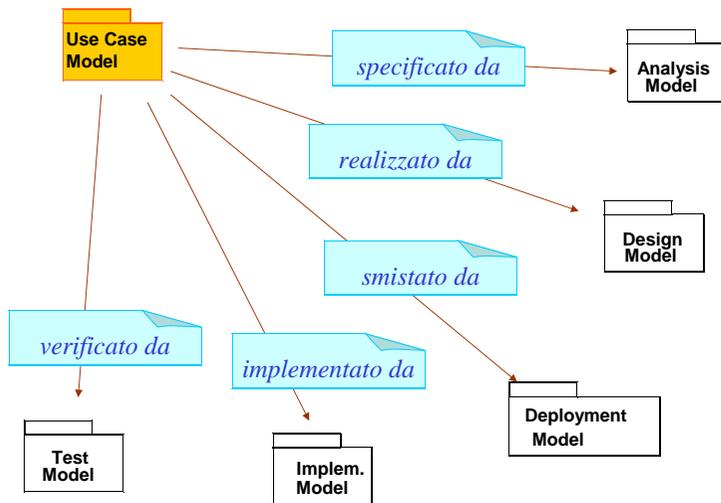
Use case diagrams



Fasi, modelli e diagrammi



Use case model e dipendenze



Use-case driven design

Lo **use case** definisce la funzionalità che è utilizzata come base per lo sviluppo delle fasi di analisi, di progetto (*disegno*), di implementazione e di test.

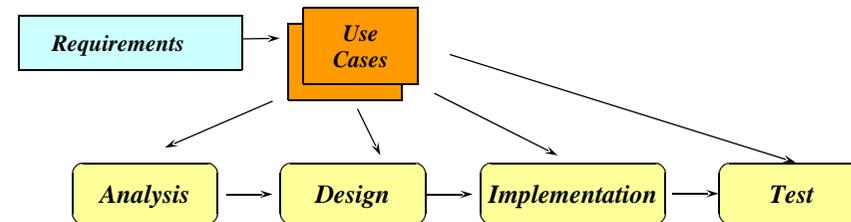


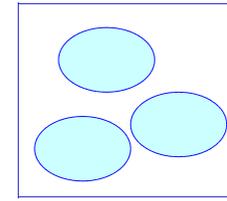
Diagramma use case

- Un diagramma use case è un grafo composto da attori, un insieme di *use case* opzionalmente racchiusi da un rettangolo, associazioni tra attori e use case, relazioni tra use case e generalizzazioni tra attori.
- Un *diagramma use case* è un dispositivo di modellazione usato per descrivere i requisiti funzionali di un sottosistema o di una semplice classe, ma anche dell'intero sistema da analizzare.
- Nel "Unified Process" un "Use Case Model" descrive completamente i requisiti funzionali dell'intero sistema da analizzare raccogliendo tutti gli use case relativi ai sottosistemi in cui è stato scomposto il sistema.

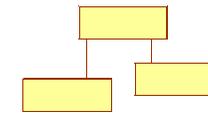
Prime fasi di utilizzo dei diagrammi Use Case

Analisi requisiti

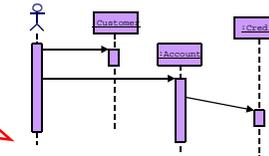
Diagrammi Use Case



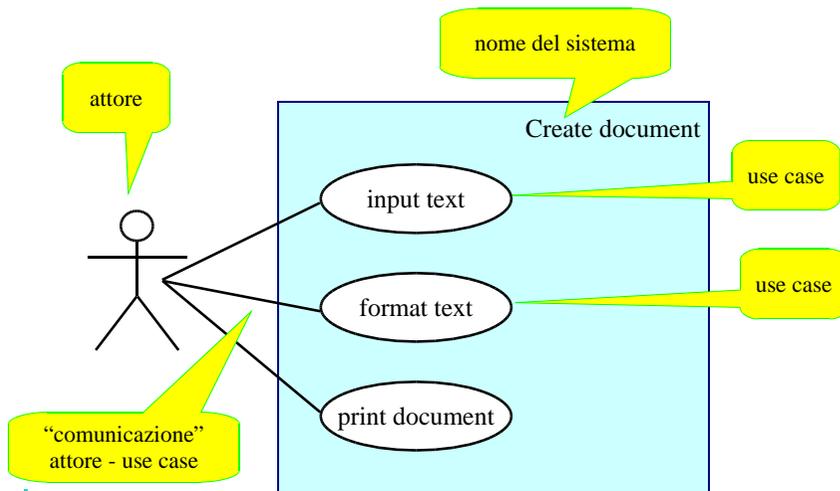
Diagrammi delle Classi



Diagrammi di interazione (comunicazione e/o sequenza)



Esempio di diagramma use case



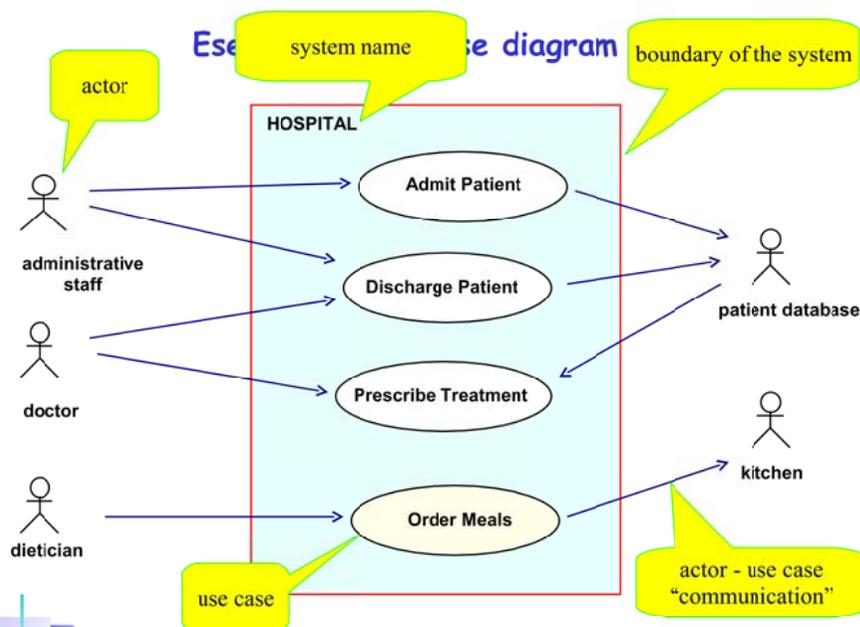
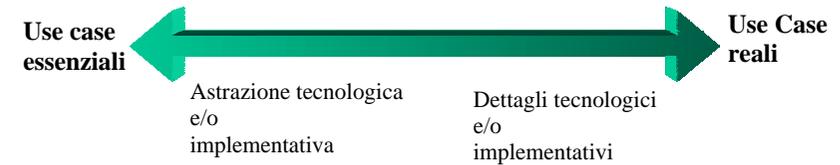
Use Case

- La funzionalità completa del sistema è rappresentata da un insieme di *use case*.
- Ciascun *use case* specifica una "funzionalità finita" del sistema (...), un servizio.
- Un diagramma *use case* è la descrizione di un insieme di azioni che un sistema compie per dare dei risultati osservabili ad un attore.

Un diagramma use case serve per...

- Specificare il contesto di un sistema
- Catturare i requisiti di un sistema
- Convalidare l'architettura di un sistema
- Guidare l'implementazione e la generazione di test
- Guidare la realizzazione del manuale utente

Tipi di Use case



USE CASE: raccomandazioni

- Gli identificatori degli use case devono utilizzare il nome di un verbo ⇒ e' un processo
- Le funzionalita' del sistema devono essere tracciabili all'interno degli use case
- Gli use case devono essere tracciabili all'interno del codice implementato
- I limiti (*boundary*) di un sistema possono essere:
 - limiti delle componenti HW e/o SW;
 - limiti dell'organizzazione in cui opera;
 - limiti del dipartimento in cui opera.

A system boundary

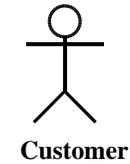
Swiss Cheese Factory

- *Determining the boundary can be a design activity.*
- *Note that the system may not be a software system; it could be a business or a hardware device for example.*

Attore (actor)

Un attore è qualcuno o "qualcosa" che interagisce col sistema.

Si rappresenta con un omino stilizzato (*stick man*), ed ha solitamente un nome



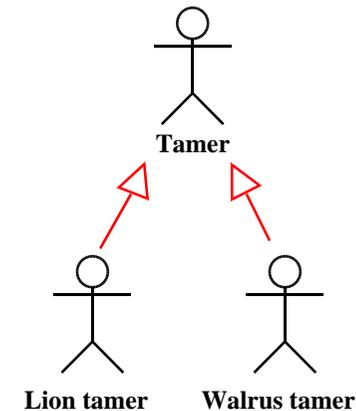
o come una classe con la specifica « actor »



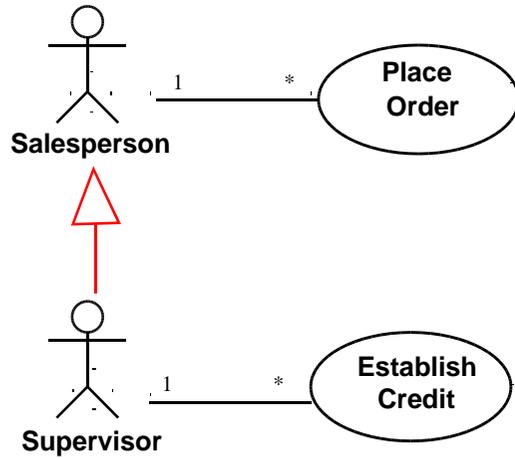
attore (2)

- Insieme coerente di ruoli che un *utente* di un caso d'uso (funzionalità) "recita" quando interagisce con esso
- Col termine "qualcosa" vogliamo intendere che un attore può essere una persona, un dispositivo od un altro sistema a se stante.
- Gli attori possono essere attivi o passivi:
 - un attore *attivo* avvia un use case;
 - un attore *passivo* semplicemente riceve informazioni da un use case.

Generalizzazione tra attori



Generalizzazione tra attori (2)

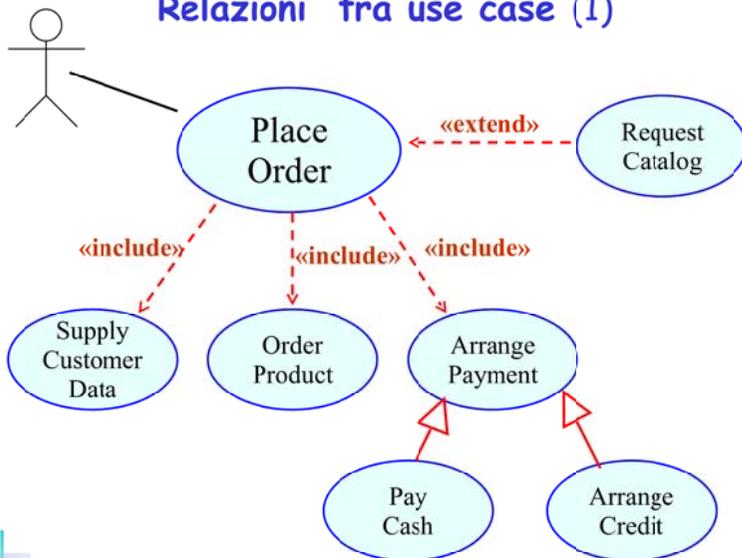


Relazione attore - use case

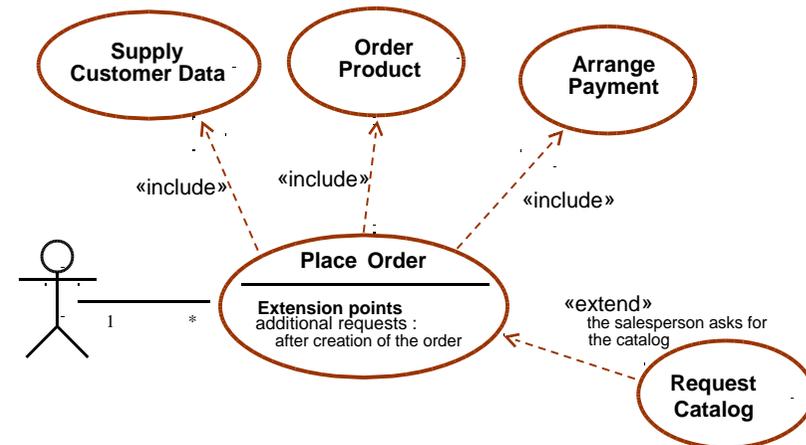


- Un attore interagisce con un use case mediante una **associazione di comunicazione** (*communication association*).
- Questa è **normalmente** una relazione uno a uno, priva di direzione.
- Le cose possono essere più complesse...

Relazioni tra use case (1)



Relazioni tra use case (2)



Relazioni tra use case (3)

«extend»

- **extend** - a dashed arrow indicating an addition to functionality of the base case.

«include»

- **include** - a dashed arrow indicating a calling relationship like a function call.



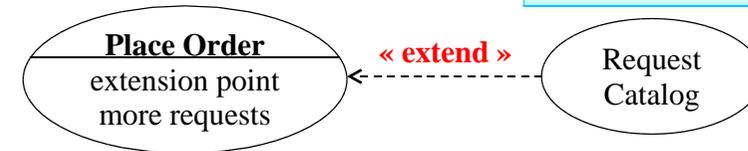
- **generalization** - a hollow arrowhead indicating

"The UML User Guide", 1999. (The Three Amigos)

« extend »

base use case

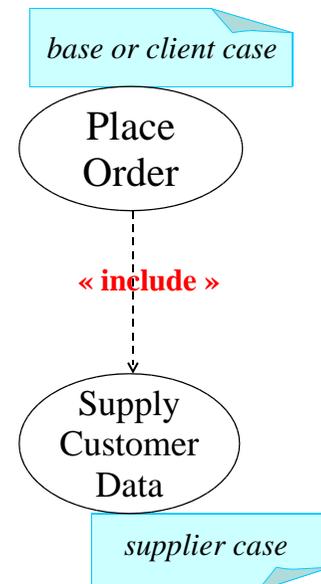
extension client



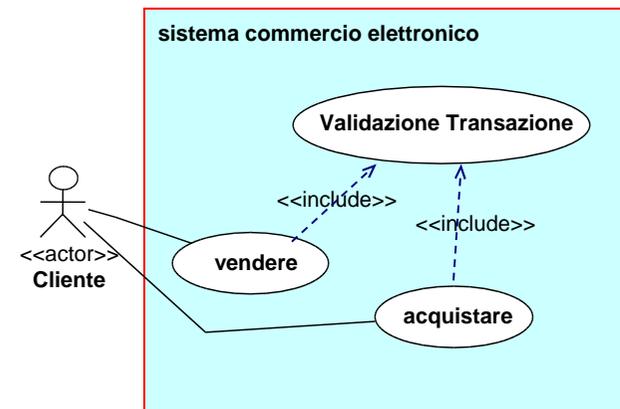
- Il *client use case* aggiunge funzionalità al *base use case*, inserendo sequenze di azioni nelle azioni di base.
- Il *base use case* ad un certo punto valuterà una condizione che, se risulta vera, l'azione "estesa" verrà inserita.
- Il *client use case* ha senso solo nel contesto del *base use case*.
- Il *base use case* implica l'estensione.

« include »

- La relazione *include* e' come una chiamata a funzione.
- Si svolge sotto il controllo del *case base (o client)*.
- Il *base case* è in esecuzione e quando richiesto è richiamato il "fornitore" (*supplier case*).
- Il *supplier use case* è un use case a se stante.



esempio include

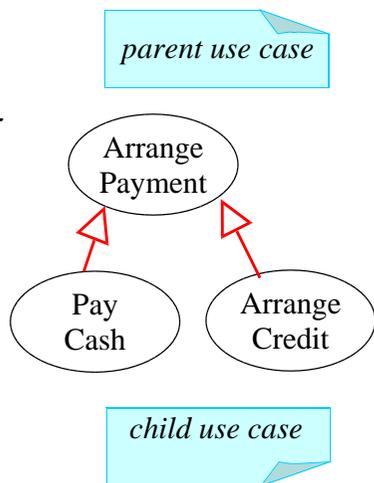


Generalizzazione

L'eredità tra use case implica la sostituibilità.

Ogniqualevolta *ArrangePayment* è utilizzabile, *PayCash* potrebbe essere accettabile.

Il *child use case* può "estendere" il comportamento di *parent use case*, tuttavia potrebbe essere presa in considerazione la sostituibilità col *parent use case*.



How to

- Establish the context of the system by identifying the actors that surround it.
- For each actor, consider the behaviour that each expects or requires the system to **provide**.
- Name these common behaviours as use cases.
- Factor common behaviour into new use cases that are used by others; factor variant behaviour into new use cases that extend more main line flows.
- Model these use cases, actors, and their relationships in a use case diagram.
- Adorn these use cases with notes that assert non functional requirements; you may have to attach some of these to the whole system.

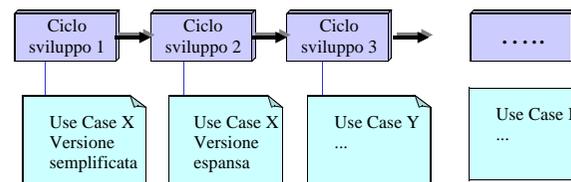
"The UML User Guide", 1999. (The Three Amigos)

Fasi per la redazione degli Use Case

1. Dalle funzionalità del sistema definire:
 - gli attori;
 - gli use case.
2. Scrivere gli use case ad alto livello
 - deciderne le prioritá (primario, secondario, opzionale)
3. Disegnare lo Use Case Diagram
4. Relazionare gli use case tra di loro
 - descrivere tali relazioni
5. Espandere gli use case piú importanti (use case espansi)
6. Rank use case

Rank Use Case

- Schedulare i cicli di sviluppo in base agli Use Case



esempio tabella di ranking

| Use Case | a | b | c | d | Somma |
|----------|---|---|---|---|-------|
| X | 4 | 5 | 2 | 0 | 11 |
| Y | 2 | 1 | 2 | 1 | 6 |
| K | 1 | 1 | 1 | 1 | 4 |

Criteria di ranking

- Importanza architetturale
- Maggior numero di funzionalita' di base
- Alto rischio
- Difficolta' di realizzazione

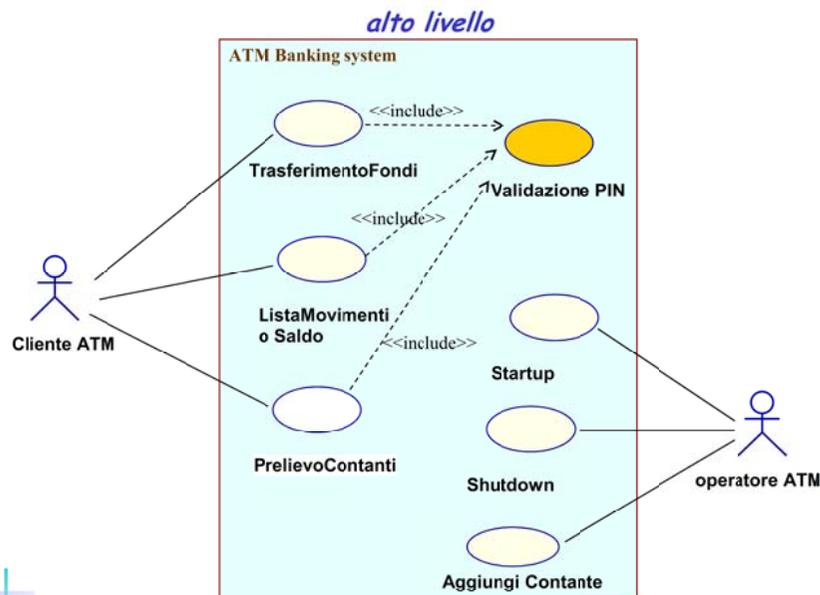
From Requirements to Analysis

- From the Use Case diagrams an initial set of objects and classes can be identified
- This is the first step of analysis
- The second step is to refine the use cases through interaction diagrams
- Class diagrams and the object oriented paradigm will be covered first

USE CASE in descrizione narrativa

- **Documento testuale** che descrive la sequenza di eventi di un attore che utilizza il sistema.
- Serve a migliorare la comprensione dei requisiti
- Passo preliminare per descrivere i requisiti del sistema
- Accompagna la descrizione grafica (diagramma use case)

Esempio " Diagramma use case sistema bancomat"



es. USE CASE "Validazione PIN" (1)

In glossario

Use case: validazione PIN (*Personal Identification Number*)

Sommario: il sistema controlla e convalida il PIN del cliente

Attore: cliente ATM

Precondizioni: il sistema è in attesa e mostra sul display un messaggio di benvenuto.

Descrizione (funzionalità):

In glossario

- 1) Il cliente inserisce l'ATM Card (*bancomat / carta di credito*);
- 2) Il sistema riconosce l'ATM Card e ne legge il numero;
- 3) Il sistema chiede il PIN;
- 4) Il cliente inserisce il PIN;
- 5) Il sistema controlla la data di scadenza e se la Card risulta in stato di smarrita o rubata;
- 6) Se la Card è valida, il sistema controlla se il PIN è corretto (PIN mantenuto dal sistema);

es. USE CASE "Validazione PIN" (2)

Descrizione (cont.):

- 7) Se il PIN è corretto, il sistema controlla quali conti correnti sono accessibili con quella Card;
- 8) Il sistema mostra al cliente le possibili transazioni (menu): prelievo, saldo, lista movimenti o trasferimento;

Alternative:

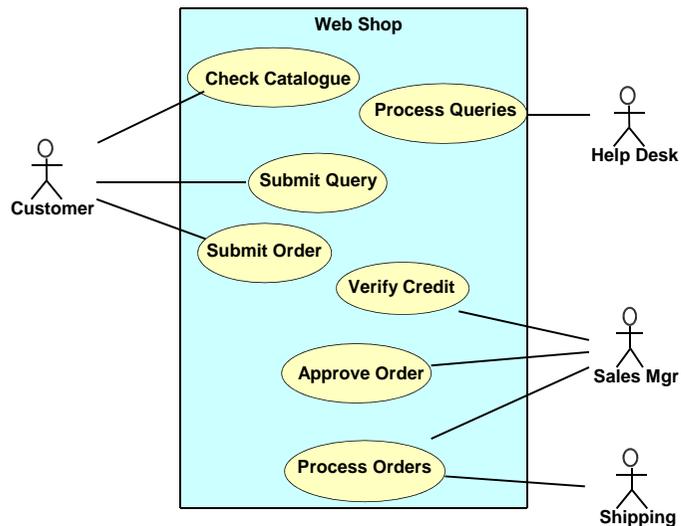
- 1) Il sistema non riconosce l'ATM Card, la Card viene espulsa;
- 2) Il sistema determina che la Card è scaduta: la Card viene confiscata;
- 3) Il sistema determina che la Card risulta smarrita o rubata: la Card viene confiscata;
- 4) Il cliente digita un PIN non corretto...
- 5) ...

Postcondizioni:

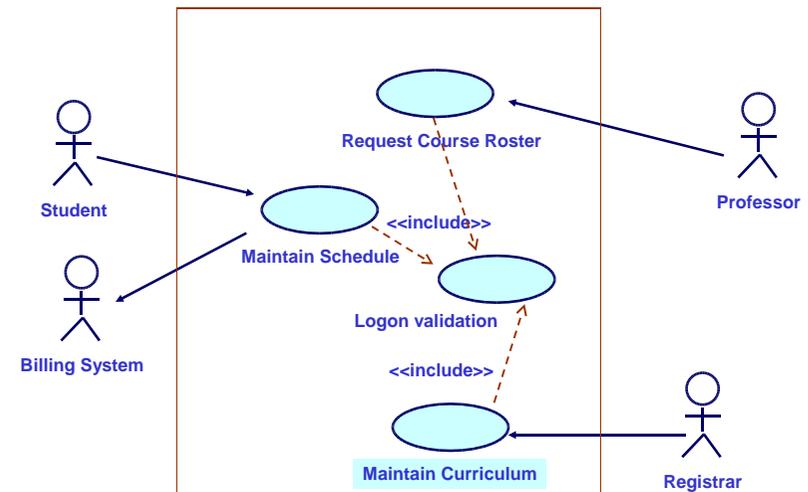
Il PIN è stato validato.

Esempi di diagrammi use case

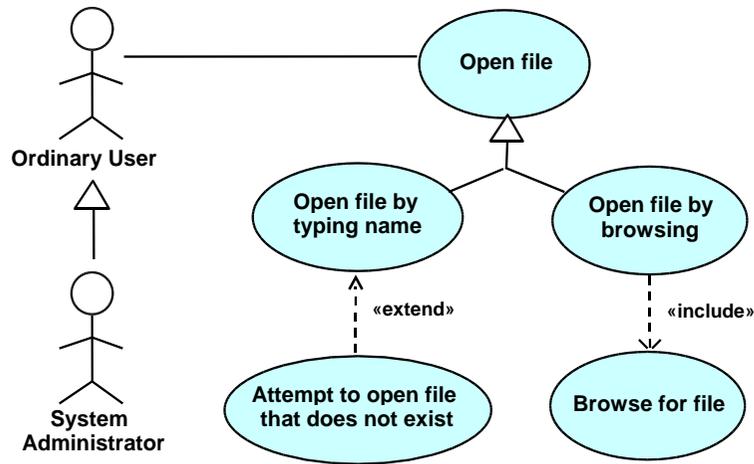
Commercio elettronico



University registration system



Apertura file



© Renato Conte - UML: USE CASE - 37 / 44 -

Example: Weather Monitoring Station

- This system shall provide automatic monitoring of various weather conditions. Specifically, it must measure:
 - wind speed and direction
 - temperature
 - barometric pressure
 - humidity
- The system shall also provide the following derived measurements:
 - wind chill
 - dew point temperature (*it. punto di rugiada*)
 - temperature trend
 - barometric pressure trend

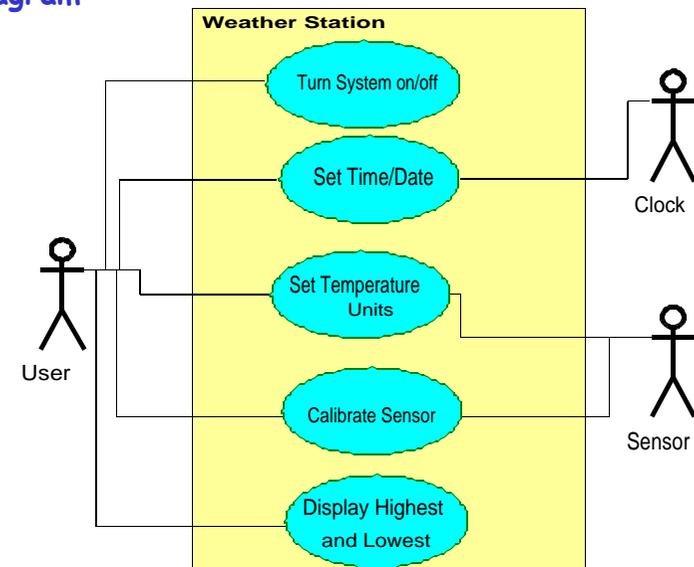
© Renato Conte - UML: USE CASE - 38 / 44 -

Weather Monitoring System Requirements

- The system shall have the means of determining the current time and date so that it can report the highest and lowest values for any of the four primary measurements during the previous 24 hour period.
- The system shall have a display that continuously indicates all eight primary and derived measurements, as well as current time and date.
- Through the use of a keypad the user may direct the system to display the 24 hour low or high of any one primary measurement, with the time of the reported value.
- The system shall allow the user to calibrate its sensors against known values, and set the current time and date.

© Renato Conte - UML: USE CASE - 39 / 44 -

Use case Diagram



© Renato Conte - UML: USE CASE - 40 / 44 -

Scenario 1: Powering Up

- 1 Power is turned on
- 2 Each sensor is constructed
- 3 User input buffer is initialized
- 4 Static elements of display are drawn
- 5 Sampling of sensors is initialized

The past high/low values of each primary measurement is set to the value and time of their first sample.

The temperature and Pressure trends are flat.

The input manager is in the Running state

and so on ...

Bibliografia

Grady Booch, James Rumbaugh, Ivar Jacobson.
The Unified Modeling Language User Guide, Addison Wesley ,
(1999).

Grady Booch, James Rumbaugh, Ivar Jacobson
The Unified Modeling Language Reference Manual, Addison
Wesley, (1999).

Ivar Jacobson, Grady Booch, James Rumbaugh
The Unified Software Development Process,
Addison Wesley, (1999).

R.S Pressman "Principi di Ingegneria del software" 4° ed
McGraw-Hill, (2004).

Riferimenti nel Web

OMG UML - Reference manual UML 1.5
www.omg.org/uml/

Materiali e riferimenti su CASI D'USO
alistair.cockburn.us/usecases/usecases.html

UML: tool, demo,doc
www.rational.com

UML: Tutorial e link
www.kobryn.com