

Studio di caso:
Bancomat
Automated Teller Machine

uso dei diagrammi:
use case,
classi,
collaborazione,
stato

Bibliografia

- Hassan Gomaa: "Designing concurrent distributed and real time applications with UML", Addison Wesley 2000
- I.Jacobson, G.Booch, J.Rumbaugh: "The Unified Software Development Process", Addison Wesley 1999
- Reference manual UML 1.4

Riferimenti nel Web

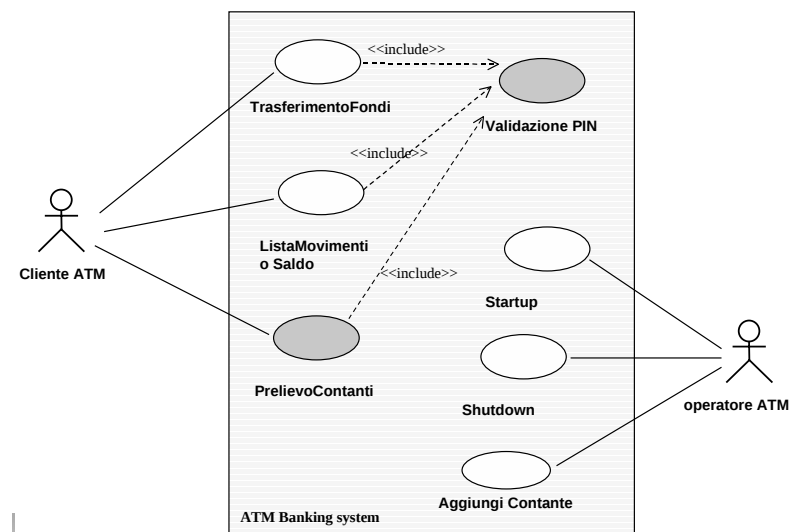
UML 1.4 RTF:
www.celigent.com/omg/umlrtf

OMG UML Tutorials:
www.celigent.com/omg/umlrtf/tutorials.htm

UML 2.0 Working Group:
www.celigent.com/omg/adptf/wgs/uml2wg.htm

OMG UML Resources:
www.omg.org/uml/

Use case sistema bancomat (ATM) - alto livello -

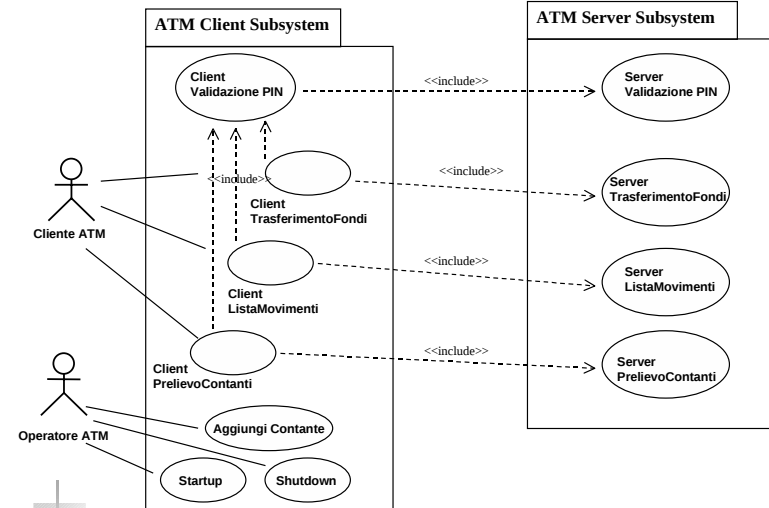




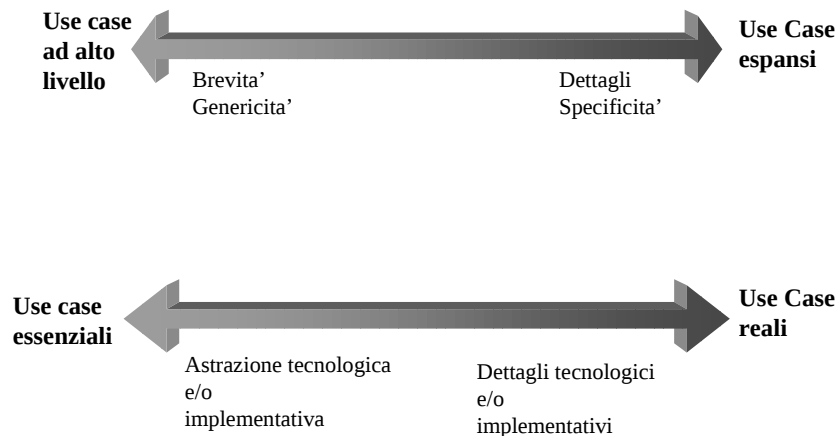
USE CASE raccomandazioni

- Gli identificatori degli use case devono utilizzare il nome di un verbo $\Rightarrow$ 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 di un sistema:
 - Limiti delle componenti HW e/o SW;
 - Limiti dell'organizzazione in cui opera
 - Limiti del dipartimento in cui opera

ATM: Use case sottosistemi Client / Server



Tipi di Use case



USE CASE in descrizione narrativa

- Documento testuale che descrive la sequenza di eventi di un attore che utilizza il sistema.
- Servono a migliorare la comprensione dei requisiti
- Passo preliminare per descrivere i requisiti del sistema

USE CASE “Validazione PIN” (1)

In glossario

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

Sommario: il sistema controlla e valida 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);

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.

USE CASE “Prelievo contanti”

Use case: Prelievo contanti

Sommario: il cliente preleva una determinata quantità di denaro da un valido conto corrente.

Attore: cliente ATM.

Dipendenze: include lo use case “Validazione del PIN”.

Precondizioni: l'ATM è in attesa mostrando un messaggio.

Descrizione:

- a) il cliente seleziona dal menu l'opzione “prelievo”, inserisce l'ammontare da prelevare, ... ;
- b) il sistema controlla ...;
- ...

Alternative:

il sistema non ha sufficienti contanti ...

Postcondizioni:

il prelievo è stato fatto.

Modello statico del sistema

diagramma delle classi



diagramma delle classi

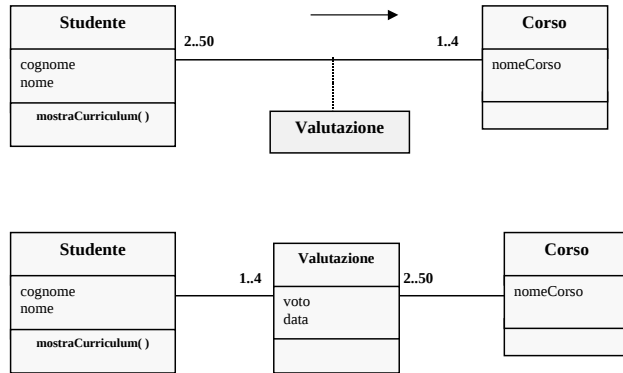
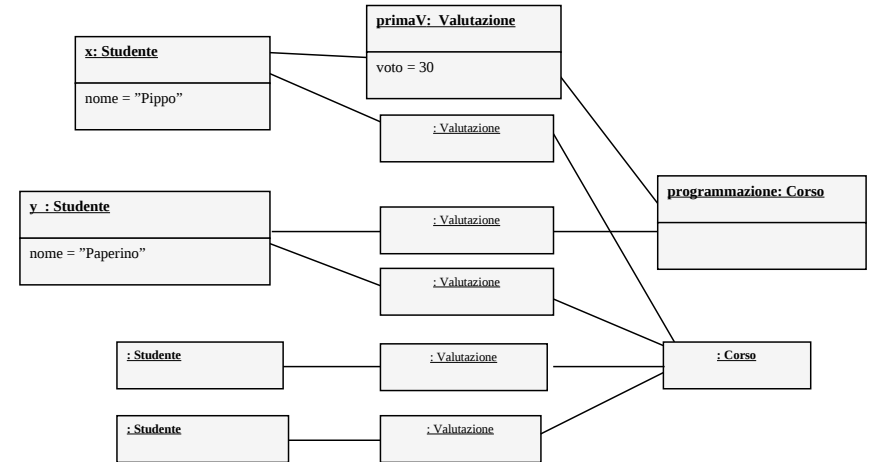
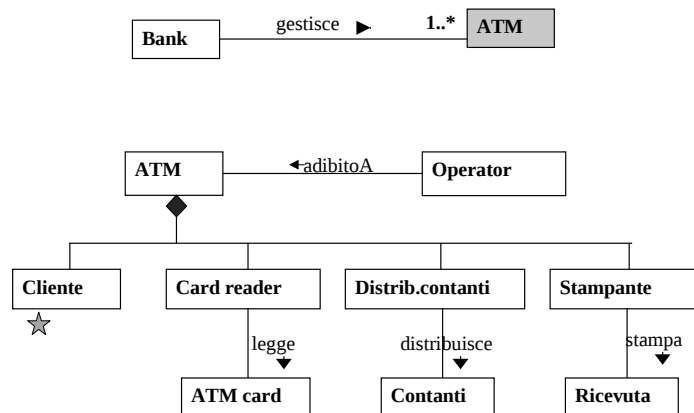


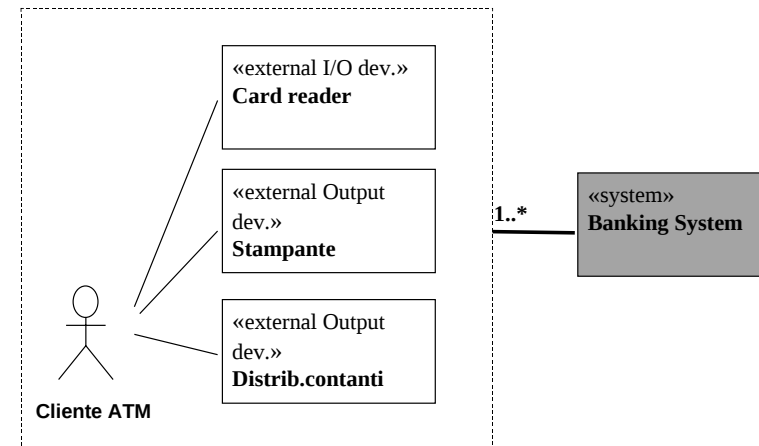
diagramma degli oggetti



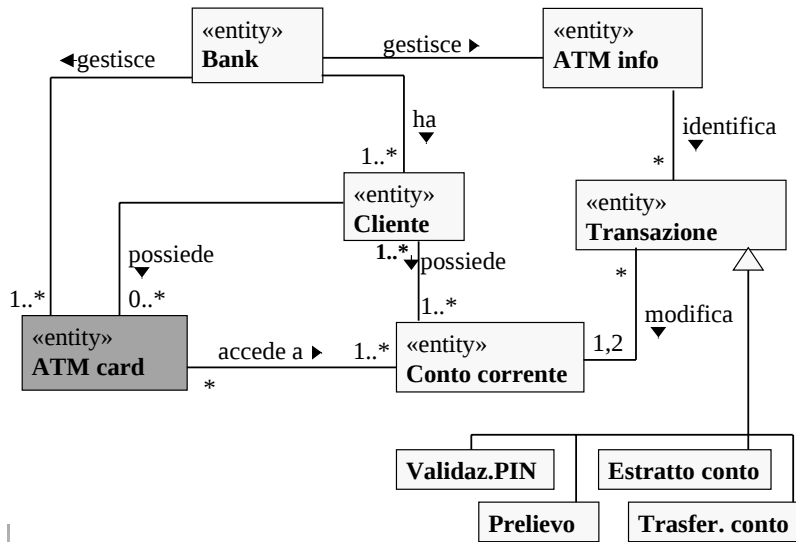
Modello statico concettuale: classi fisiche



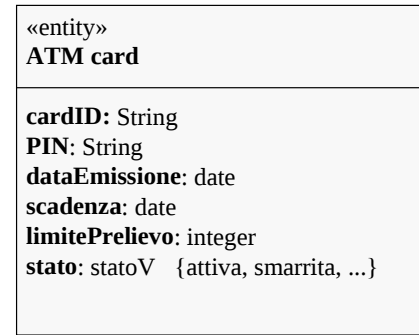
Modello statico del contesto: classi external, system



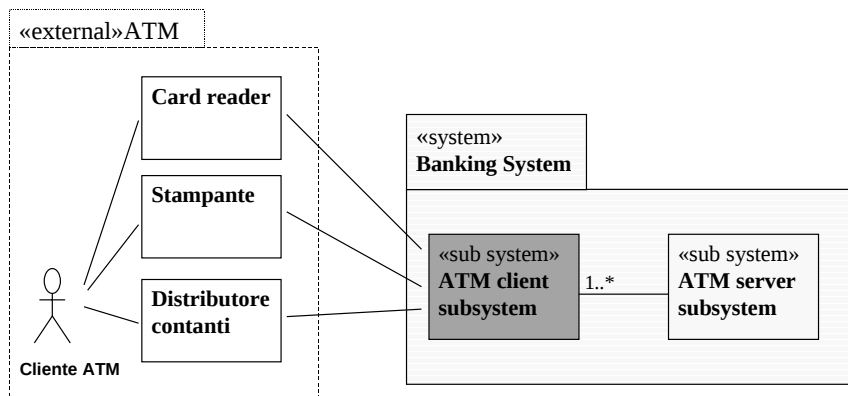
Modello statico: Banking System



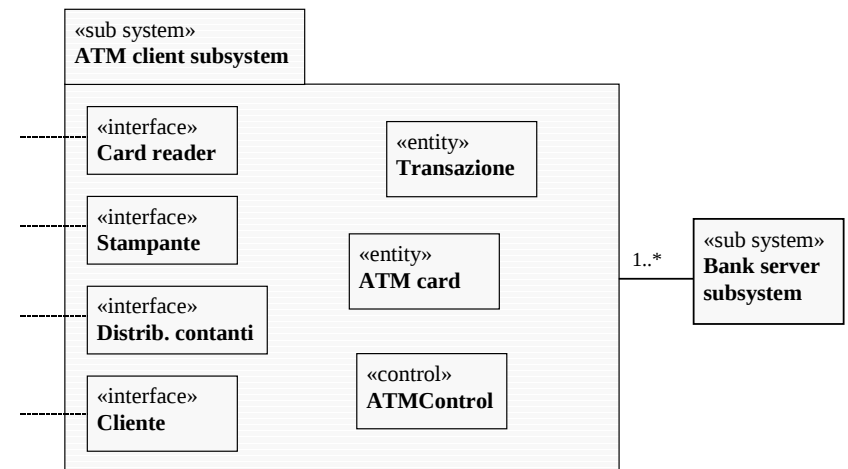
Dettagli per la classe ATM card (attributi)



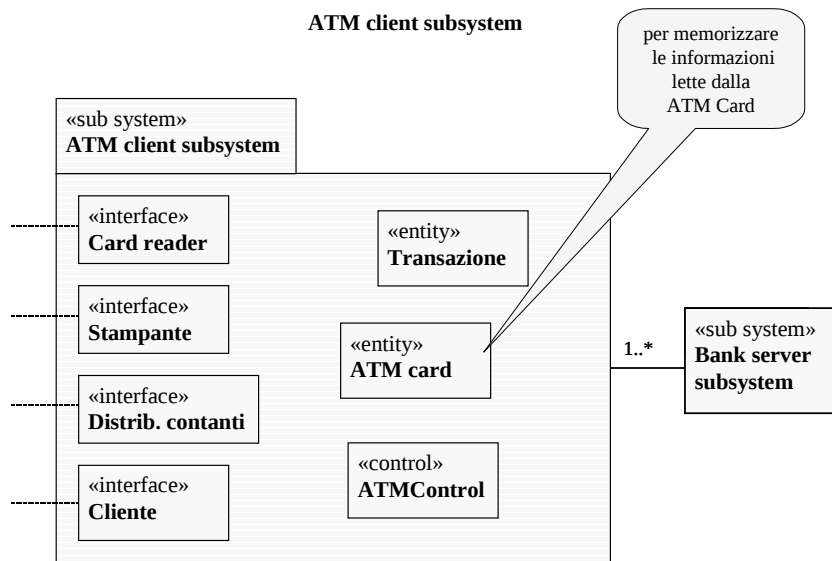
Banking system: applicazione client/server



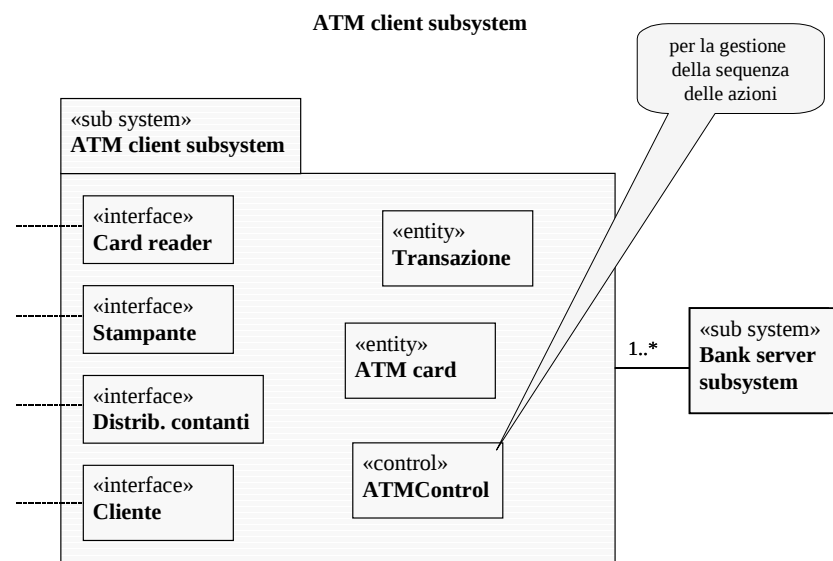
ATM client subsystem



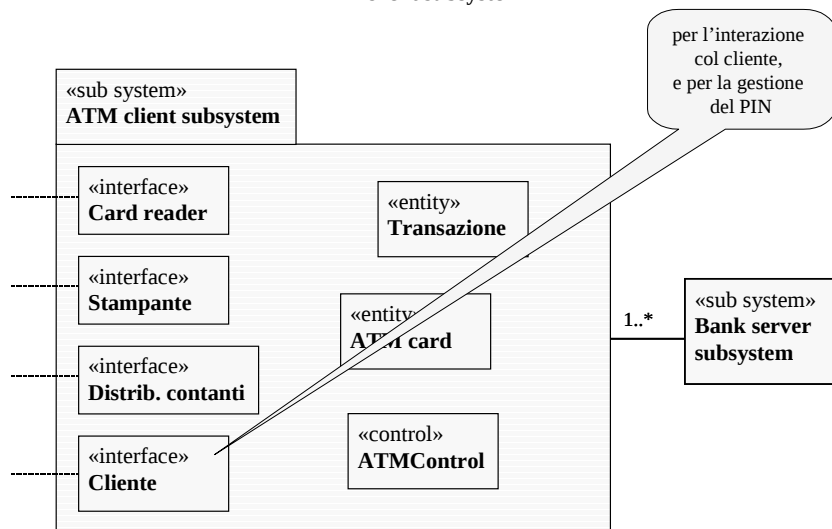
ATM client subsystem



ATM client subsystem

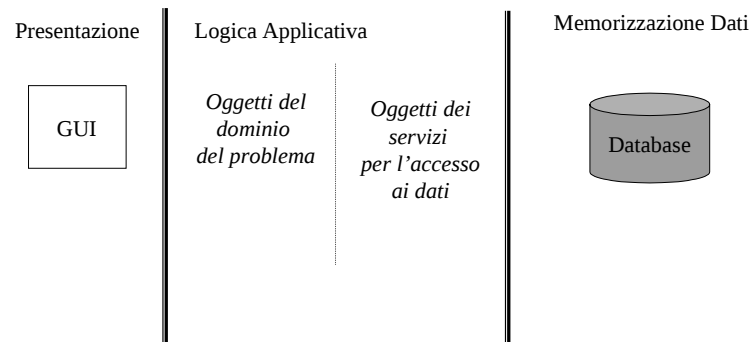


ATM client subsystem



Livelli Architeturali

- Architettura multi-layer (multi-tier)



Modellazione dinamica

Dynamic Modeling



Diagrammi dinamici

- Interazioni tra oggetti che partecipano in ciascun “use case”
- Sequenza di messaggi intra-oggetti
- Diagrammi di sequenza e/o di collaborazione e di stato
(*sequence, collaboration, statechart diagrams*)



Diagramma di collaborazione

Collaboration Diagrams

- Un diagramma che illustra le interazioni tra oggetti organizzandole attorno agli oggetti e ai link tra questi oggetti.
- A differenza di un diagramma di sequenza, evidenzia le relazioni tra gli oggetti.

Diagramma di collaborazione: use case “Validazione PIN”

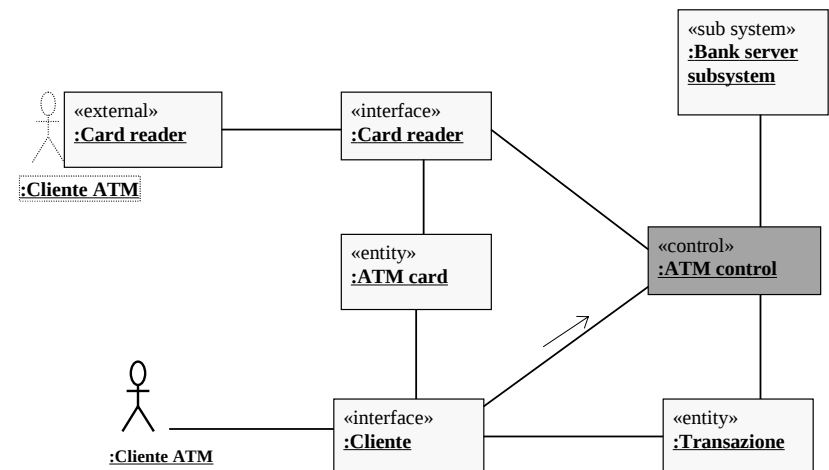
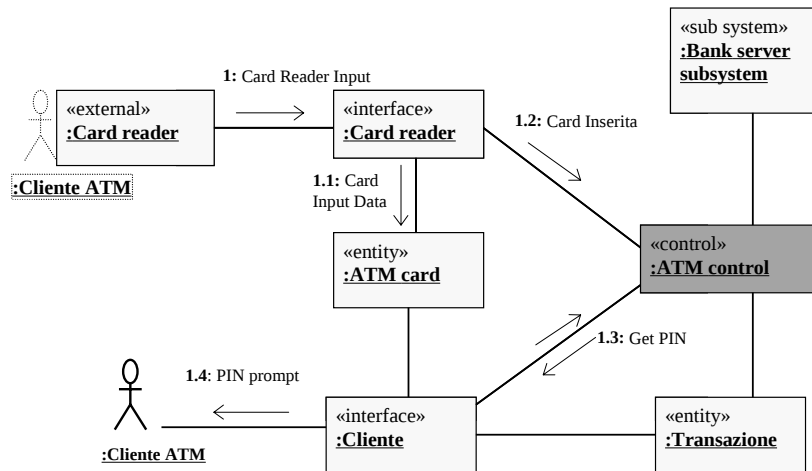
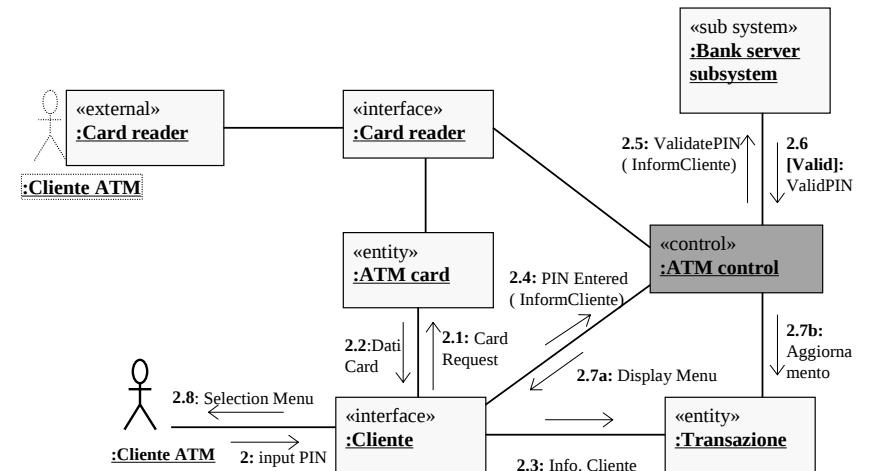


Diagramma di collaborazione: use case “Validazione PIN”



© Renato Conte - UML: Studio di caso - 25 -

Diagramma di collaborazione: use case “Validazione PIN”

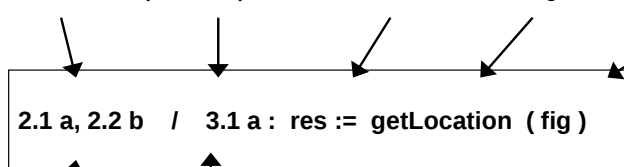


© Renato Conte - UML: Studio di caso - 25 -



Diagramma di collaborazione: arrow label

Predecessor / sequence-expression: return-value := message-name (argument-list)



Progressione dei messaggi (livello di annidamento della procedura). Sequenza di interi separati dal punto: numerazione decimale di Dewey.

Lettere dell'alfabeto: nome thread.

3.7.4: move (5, 7)

Lista di numeri di sequenze di altri messaggi

Clausola opzionale. Indica che il messaggio non sarà attivato fino a che tutti i messaggi indicati nella lista non avranno terminato le loro azioni (sincronizzazioni).

© Renato Conte - UML: Studio di caso - 26 -



Diagramma di collaborazione: arrow label

Pred. / seq.exp: retval := mess(arg.-list)

3.1.2 [Z > 0] : res := sadf (1,2,3)

Opzionale. La **clausola condizionale** è normalmente espressa in pseudo codice ed esprime quali condizioni devono verificarsi per la attivazione del messaggio.

1.2.5 [free memory] : createCopy ()

© Renato Conte - UML: Studio di caso - 27 -



Diagramma di collaborazione: arrow label

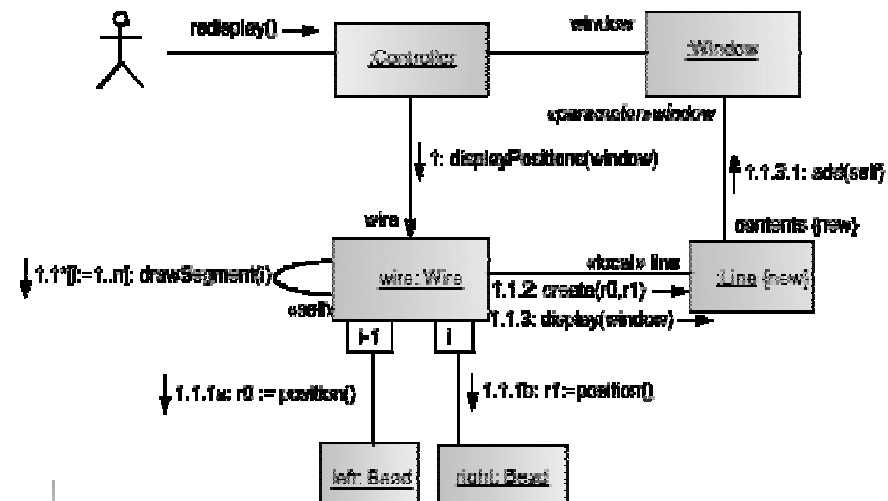
Pred. / seq.exp: retval := mess(arg.-list)

3.1.2 *[i:= 1..N] : drawSegment (i)

Opzionale. La **clausola iterativa** è normalmente espressa in pseudo codice ed esprime una esecuzione iterativa.



Esempio di diagramma di collaborazione



Descrizione dei messaggi: use case “Validazione PIN” (1)

- 1: Card Reader Input --- il ClienteATM inserisce la Card: la CardReader legge i dati.
- 1.1: Card Input Data --- la CardReader manda i dati della Card (ID, scadenza,...) all'oggetto ATMCard che li memorizza.
- 1.2: Card Inserita --- la CardReader invia l'evento “Card Inserita” all' ATM control. Come risultato vedremo che nel diagramma di stato relativo all' ATM control, un cambiamento di stato da “Idle”(inattivo) ad “Attesa PIN”.
- L'evento associato all'uscita è “Get PIN”.
- 1.3: Get PIN --- l' ATM control manda l'evento “Get PIN” all'oggetto interfaccia Cliente.
- 1.4: PIN prompt --- l' interfaccia Cliente manda la richiesta di inserimento PIN (PIN Prompt) all'attore Cliente ATM.

Descrizione dei messaggi: use case “Validazione PIN” (2)

- 2: input PIN --- il Cliente ATM manda il PIN all'oggetto interfaccia Cliente.
- 2.1: Card Request ---
- 2.2: Dati Card ---
- 2.3: Info. Cliente ---
- 2.4: PIN Entered (InformCliente) ---
- 2.5: ValidatePIN (InformCliente) ---
- 2.6 [Valid]: Valid PIN --- Il Bank Server valida il PIN e manda il segnale “validPIN” all' ATM control. Come risultato di questo evento l' ATM control passa allo stato di “Attesa scelta operazione”
- 2.7a: Display Menu ---
- 2.7b: Aggiornamento ---
- 2.8: Selection Menu ---

...alternative...



Diagrammi di sequenza (o sequenziali)

Sequence Diagrams

Use case

➔ Interazione attore sistema

➔ Genera eventi che richiedono operazioni da parte del sistema

- per un dato use case il diagramma sequenziale e' uno schema che rappresenta:
 - gli eventi generati dagli attori
 - l'ordine degli eventi generati
 - gli eventi che intercorrono tra i sistemi
- la sequenza degli eventi segue il corso del use case (incluse le alternative possibili)
- gli eventi possono includere dei parametri

Diagramma di sequenza: use case "Validazione PIN"

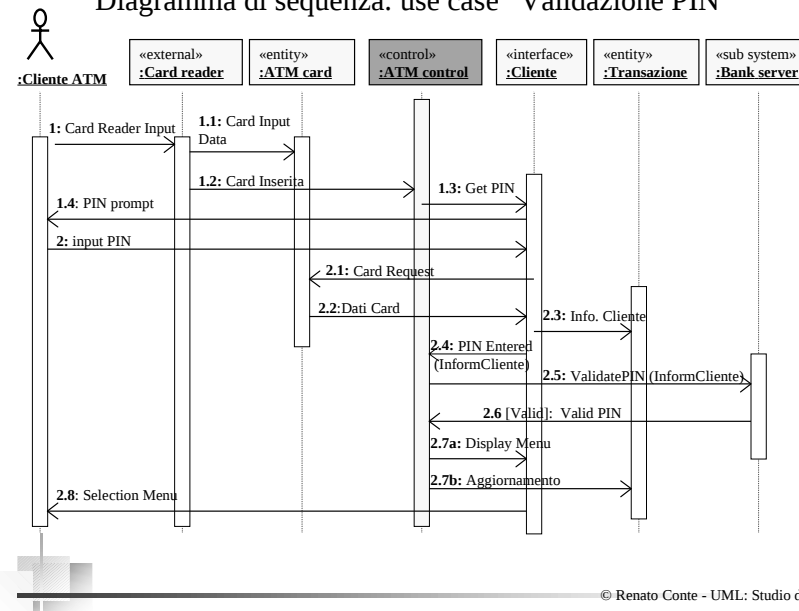


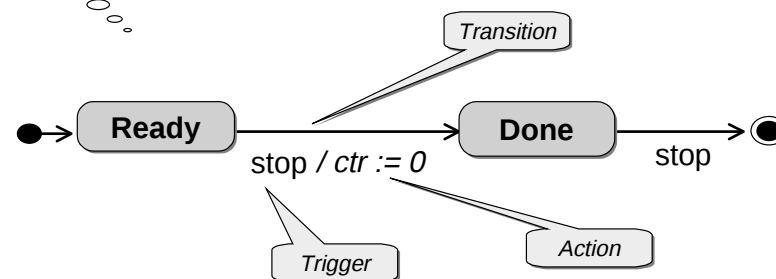
Diagramma di stato

Statechart Diagram

- Specifica il ciclo di vita degli oggetti di una classe, definendo le regole che lo governano.
- Quando un oggetto si trova in un certo stato può essere interessato da determinati eventi
- Come risultato di un evento l'oggetto può passare ad un nuovo stato (transizione)
- Utilizzato in situazioni dove le transizioni di stato sono provocate da eventi asincroni.

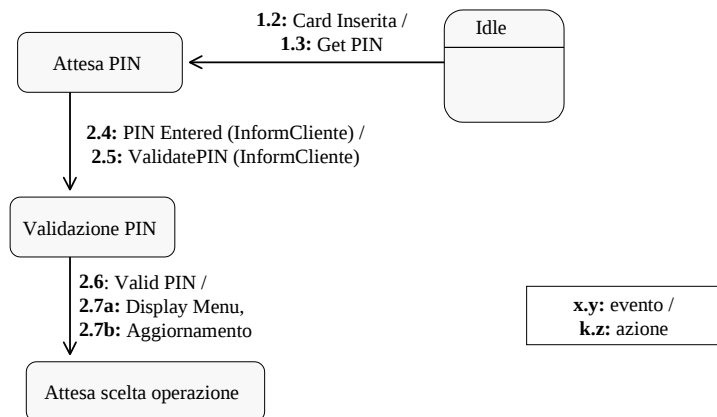


Diagramma di stato: transizione

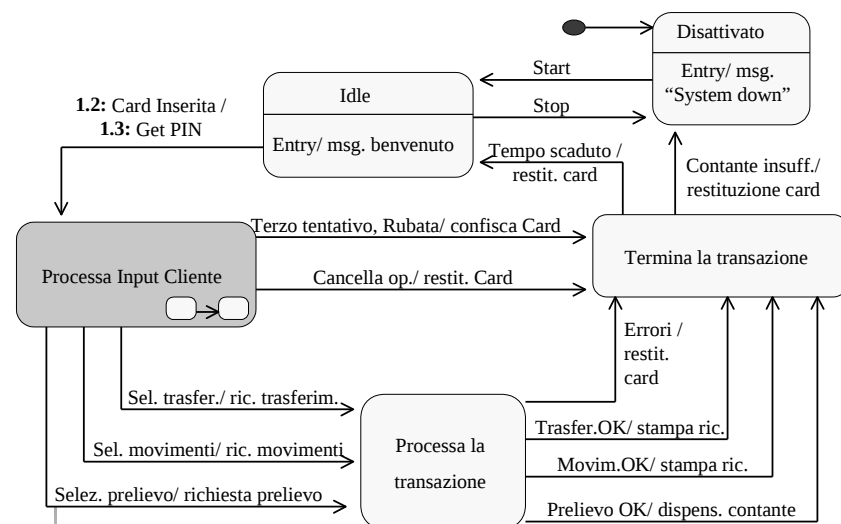


- Una transizione collega tra loro due stati
- L'uscita da uno stato definisce la risposta dell'oggetto all'occorrenza di un evento (*Trigger*)
- Una transizione è associata a un evento, una condizione (opzionale), un'azione (opzionale), e uno stato di arrivo.
- Sintassi: evento [condizione] / azione

ATM control : diagramma di stato del use case “Validazione PIN”



ATM control: diagramma di stato generale (top level)



ATM control: diagramma di stato “Processa Input Cliente”

