

CTTE

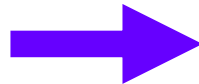
## Concur Task Trees Environment (Parte seconda)

Lezione del Corso Interazione Uomo Macchina 1, Docente Francesco Mele  
Corso di Laurea in Informatica Università di Napoli Federico II,  
Anno Accademico 2004-2005

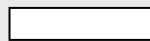
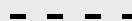
# Task e presentazioni

Il problema:

Modello del task



Presentazione 1

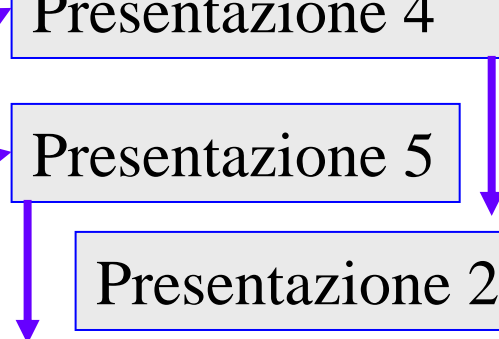
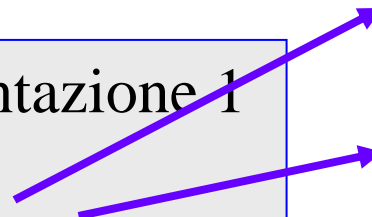


Presentazione 4

Presentazione 5

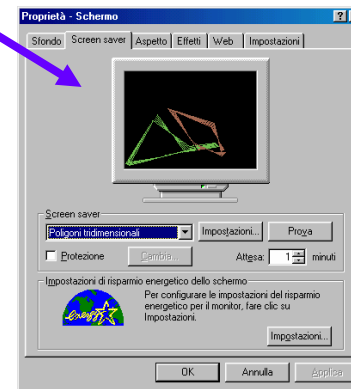
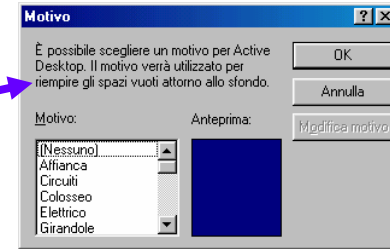
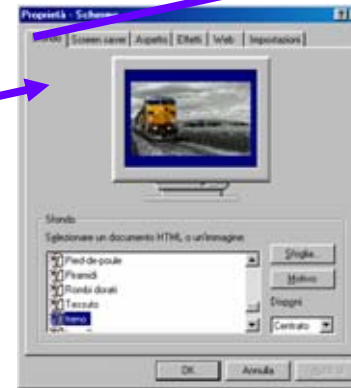
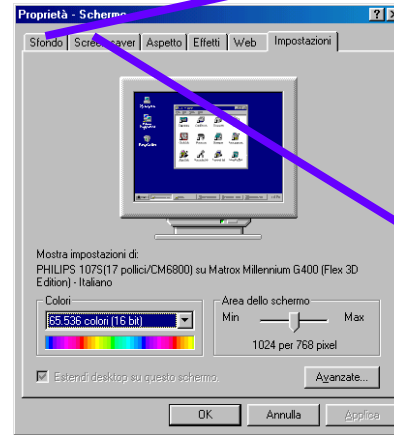
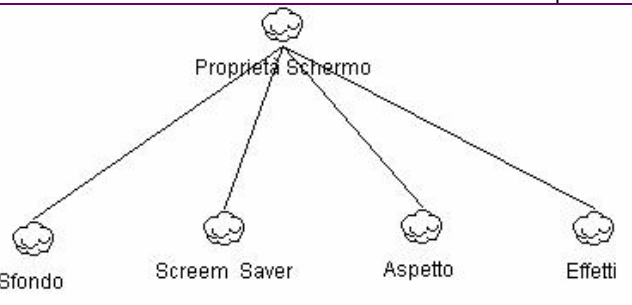
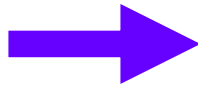
Presentazione 2

Presentazione 3



# Esemplificando su un'interfaccia nota

Modello del task



Raggruppamento delle presentazioni

Un concetto utile al problema: l'insieme dei task attivabili (ETS: Enabled Task Set) (altre traduzioni: consentiti, permessi, abilitati)

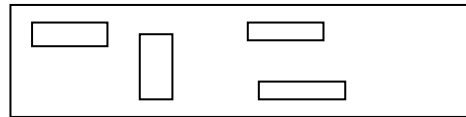
Un insieme di task attivabili è un insieme di task che sono logicamente attivabili a partire da uno stesso istante di tempo

Due prime conseguenze immediate:

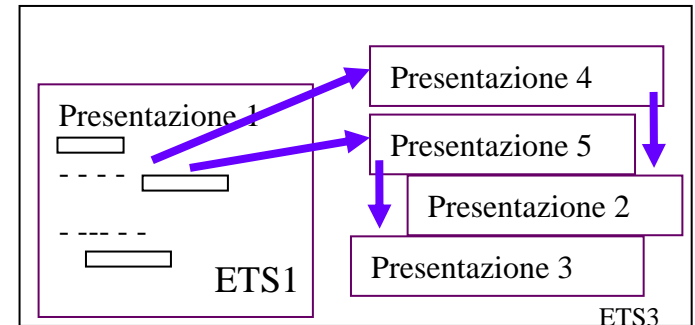
1. un task può appartenere a più ETS;
2. se dei task multipli appartengono allo stesso ETS, allora essi possono essere eseguiti allo stesso tempo e quindi appartengono alla stessa presentazione (layout).

# Le possibili soluzioni per il problema delle presentazioni

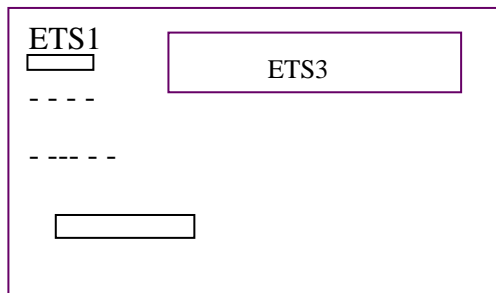
1- Un'unica presentazione per tutti i task (genera confusione e va bene per interfacce semplici)



2- Una presentazione per ogni ETS



3- Una soluzione intermedia (conservare il contesto: lasciare traccia ma disabilitare disabilitando)



Caso tipico dei task sequenziali con passaggio informazione

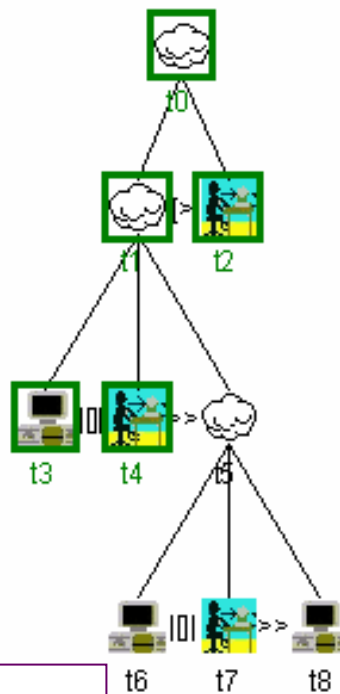
## Regole per identificare gli insiemi dei task attivabili

- I task indipendenti concorrenti  $|||$  e i task concorrenti comunicanti  $|\square|$  appartengono allo stesso ETS;
- I task sequenziali ( $T1 \gg T2$ ,  $T1 \square \gg T2$ ) dove  $T1$  abilita  $T2$ , appartengono a due differenti ETS;
- I task di scelta  $\square$  appartengono a differenti ETS, eccetto le loro prime azioni di attivazione che appartengono allo stesso task;
- Per la disabilitazione  $T1 [ > T2$   $T1$  appartiene allo stesso ETS di  $T2$ , mentre il body di  $T2$  (se esiste) appartiene ad un altro ETS

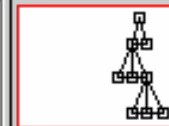


File Edit View Info Insert Tools Help

task model



Overview



Enabled Tasks

To perform a task double-click on the name in the list of enabled tasks.

Enabled Tasks



Initial State

Commands

Back

Forward

New Scenario

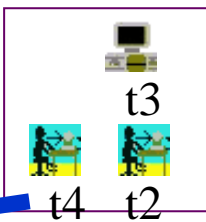
Scenario Performed

Load Scenario

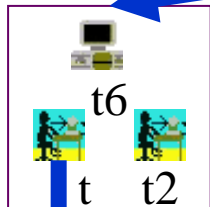
Stop Simulator

Presentazione

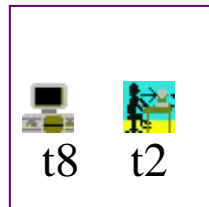
frame fo



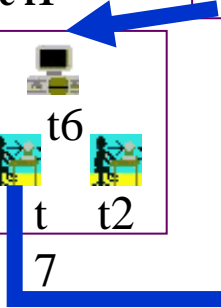
frame f1



frame f2

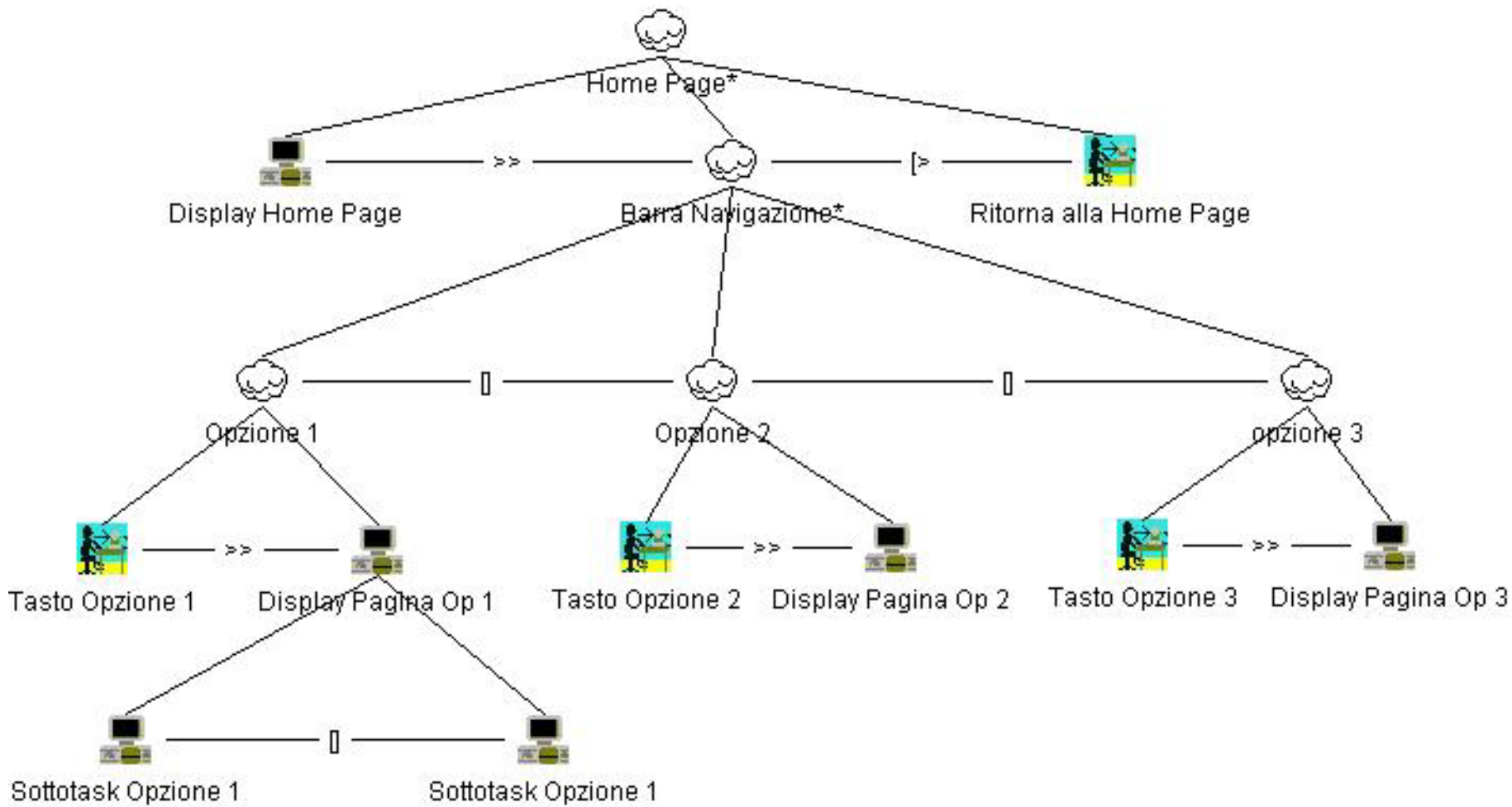


7



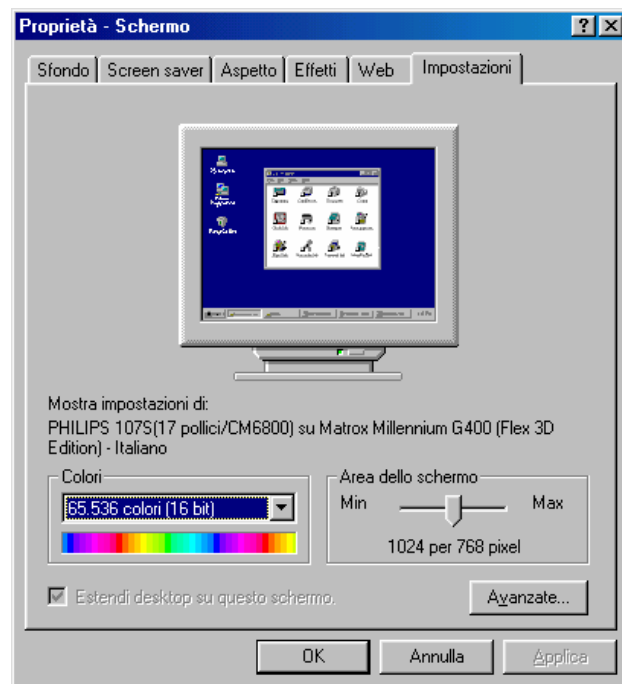


# Una soluzione (quasi) ottimale al problema barra di navigazione



# Due esercizi

- 1 Costruire l'interfaccia del task File Managing;
- 2 Verificare che l'interfaccia “proprietà schermo” sia in accordo con le regole date per verificare l'insieme degli ETS.



# Esercizi per un forum

Scelta di un'interfaccia in Internet si costruisce il relativo task model (se l'interfaccia è grande - sola una parte)

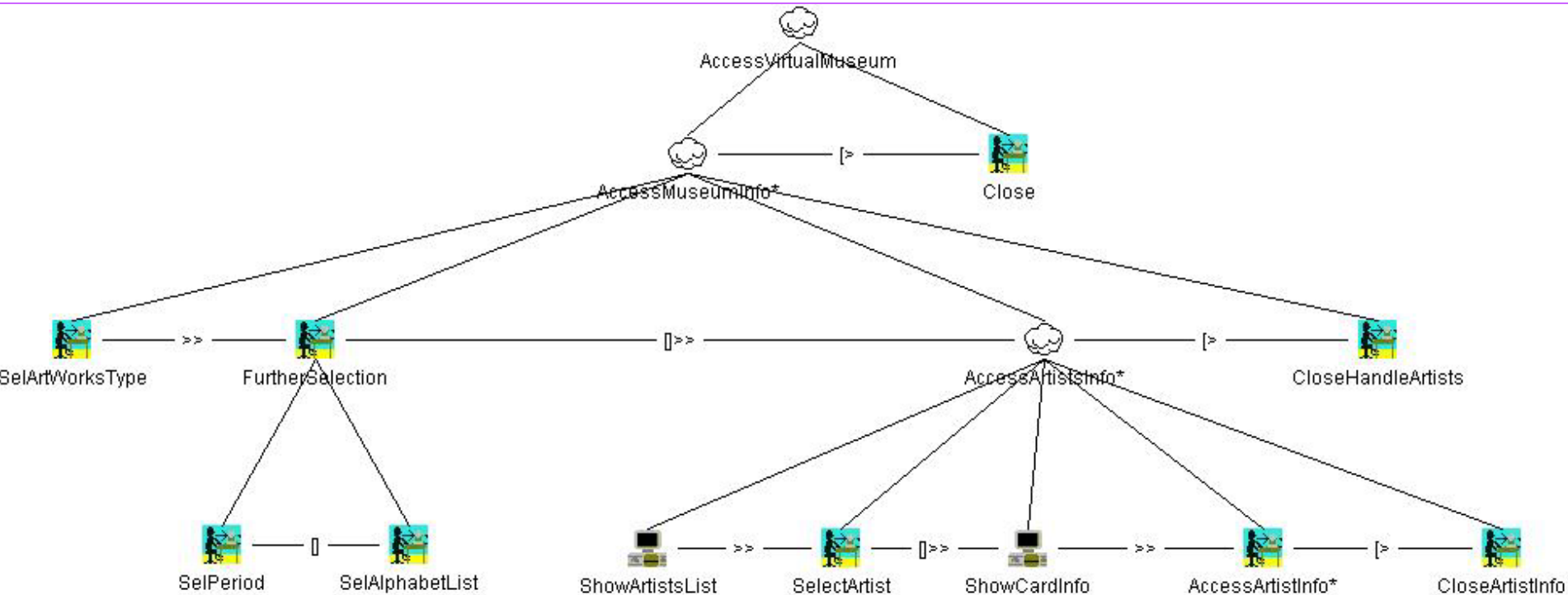
**Ogni gruppo discuterà in aula:**

- la coerenza del proprio task model con l'interfaccia scelta;
- la coerenza della simulazione con il CTTE con il reale funzionamento dell'interfaccia.

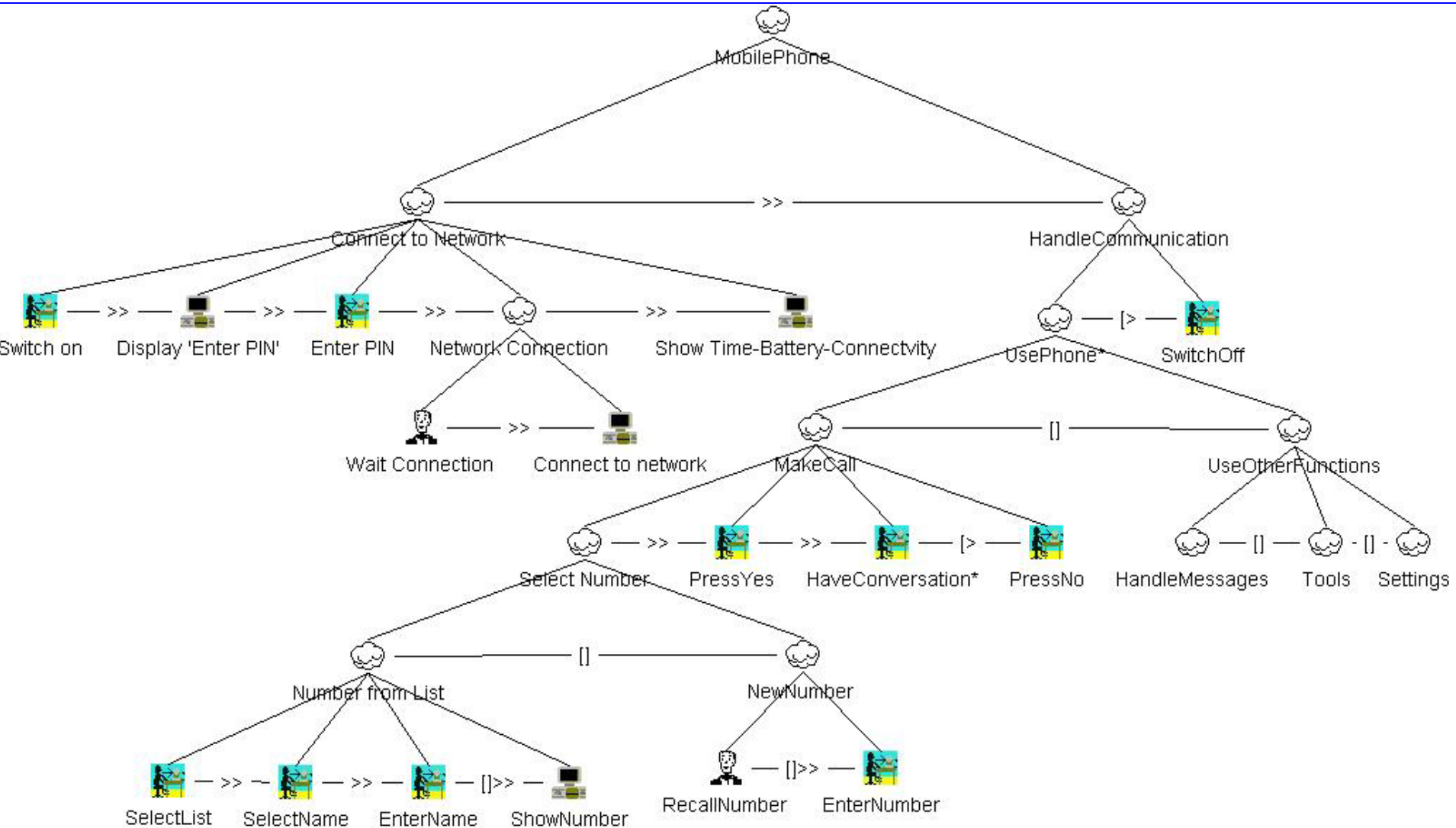
Qualche task tree da discutere insieme adesso



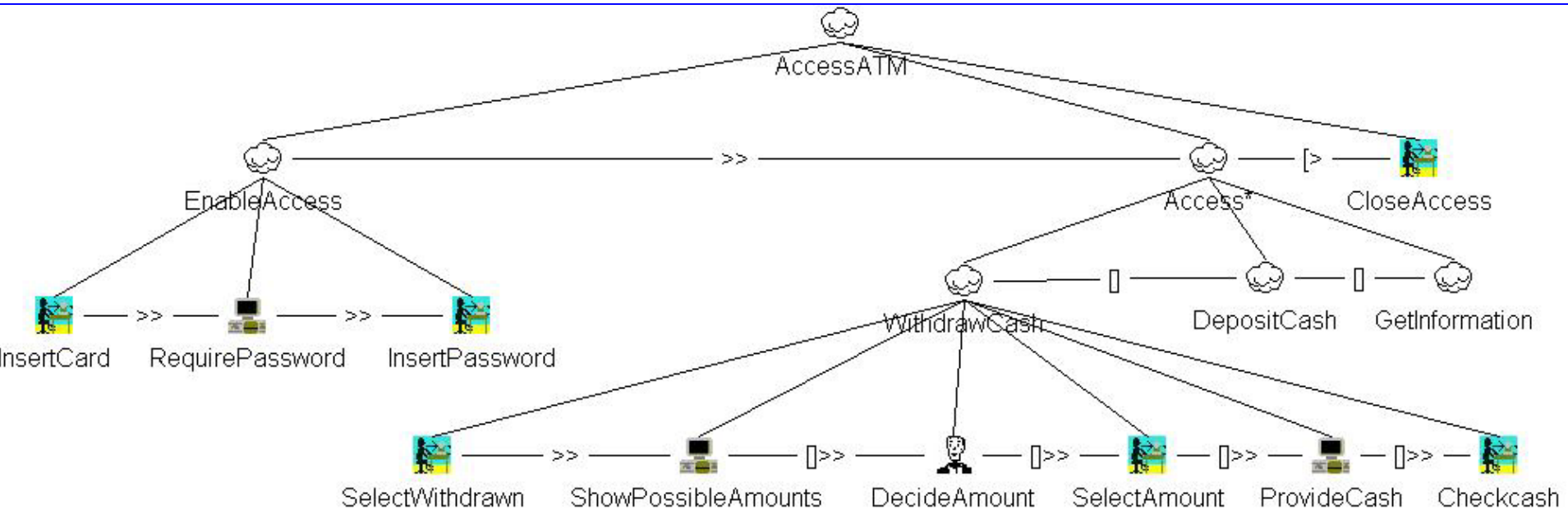
# Dal task model costruire l'interfaccia: interfaccia per accedere ad un museo



# Dal task model costruire l'interfaccia: interfaccia cellulare



# Dal task model costruire l'interfaccia: interfaccia bancomat



Dal Task Model all'architettura dell'Interfaccia:  
solo elementi teorici (poi ne riparleremo)



## Dal Task Model all'architettura dell'Interfaccia

Nel derivare un modello dell'architettura ci sono alcuni importanti elementi:

- una precisa associazione tra task e oggetti software i quali attivati dall'utente saranno eseguiti;
- l'architettura deve essere consistente con le stesse relazioni temporali stabilite nel task model;
- gli stessi effetti “semantici” descritti a livello del task devono esser presenti a livello architetturale. Per questo fine è importante allocare gli oggetti logici, identificati come necessari per eseguire il task, per gli interattori usati a livello architetturale.

# La trasformazione dal Task Model all'architettura dell'Interfaccia

Il processo di trasformazione:

- Compiti (Tasks) --> Sottocompiti
- Task --> Oggetti
- Oggetto -> Azioni

## Elementi chiave per la trasformazione

- l'identificazione degli interattori e l'associazione di ciascuno di essi con uno o più oggetti richiesti per eseguire il task;
- l'identificazione delle connessioni tra gli interattori deve permettere che il flusso dell'informazione sia tale da rendere possibile l'esecuzione del task;
- l'architettura potrebbe essere completata con le relazioni temporali definite a livelli di task.

Task name	Object name	Type	Input Action	OutputAction
<b>AccessMuseumInfo</b>	Pres_in_sel	Perc.	Inf. from User	Inf. to Museum_info
	Pres_out_sel	Perc.	Inf. from Museum_info	Inf. to User
	Museum_info	Appl.	Inf. from Pres_in_sel	Inf. to Pres_out_sel
<b>Close</b>	Pres_close	Perc.	Inf. from User	
<b>SelArtWorksType</b>	Pres_type	Perc.	Inf. from User	Inf. to Req_info
	Req_info	Appl.	Inf.fromPres_type/ Pres_list/Pres_artis t/Pres_selcri	Inf. to Museum_info
<b>FurtherSelection</b>	Pres_selcri	Perc.	Inf. from User	Inf. to Req_info
	Req_info	Appl.	Inf.fromPres_type/ Pres_list/Pres_artis t/Pres_selcri	Inf. to Museum_Info
<b>AccessArtistsInfo</b>	Museum_Info	Appl.	Inf. from Req_info	Inf. to Pres_list/ Pres_artist
	Pres_list	Perc.	Inf. from Museum_info/ User	Inf. to User/Req_info
	Req_info	Appl.	Inf.fromPres_type/ Pres_list/Pres_artis t/Pres_selcri	Inf. to Museum_info
<b>CloseHandleArtists</b>	Back_sel	Perc.	Inf. from User	

