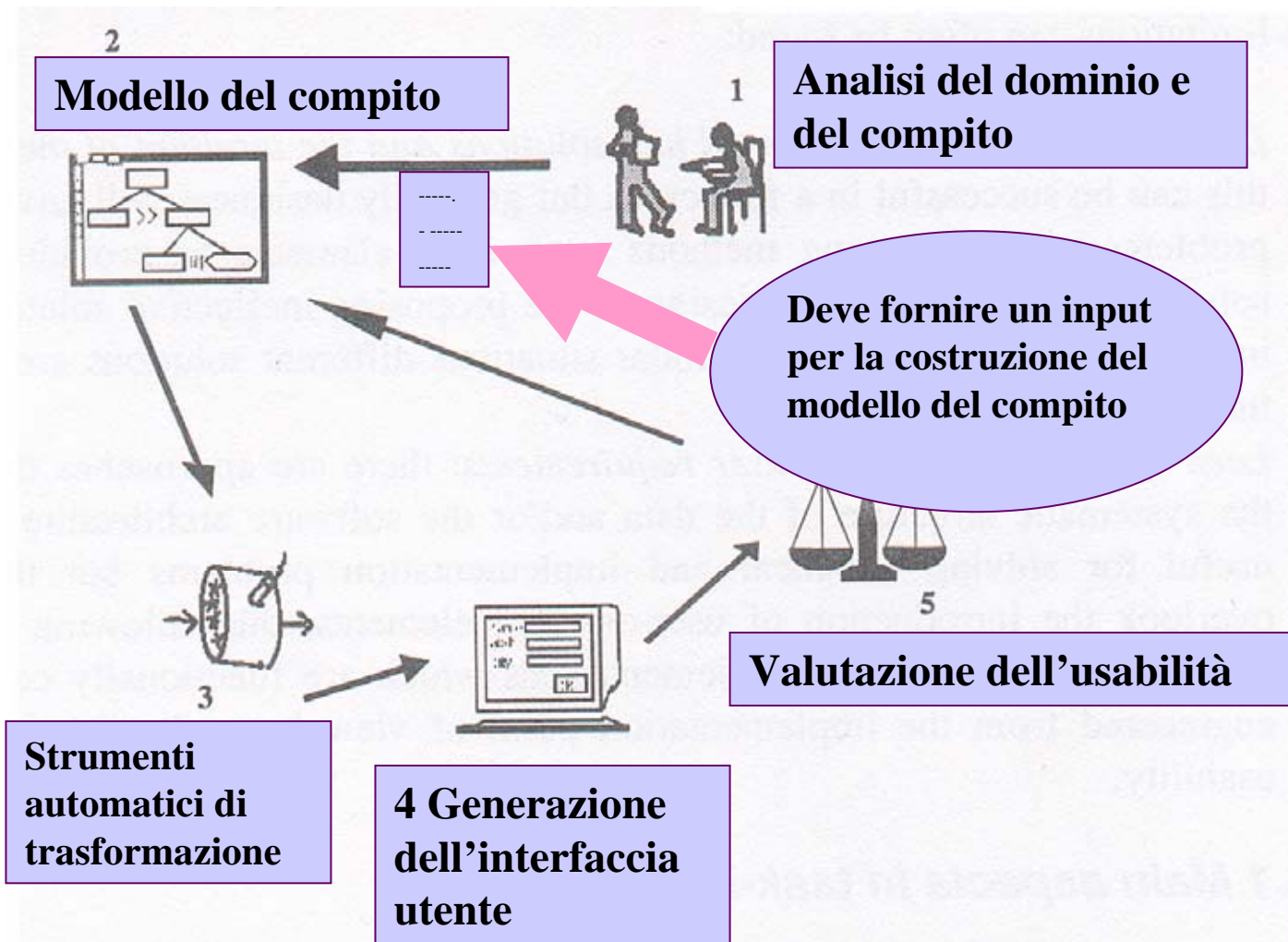


CTTE

Concur Task Trees Environment (Parte prima)

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

Ciclo di progettazione per applicazioni interattive



LO SCENARIO

Fornisce una descrizione informale di una serie di eventi in uno specifico contesto situazionale

Un primo semplice scenario:

Franco annota gli appuntamenti sulla sua agendina tascabile, ma non la consulta spesso. Franco dimentica spesso l'agenda sulla scrivania del suo studio, cerca di avere rispetto per le persone e non vuole mancare agli impegni presi (azione e tempo stabilito)

Un secondo scenario:

Franco annota gli appuntamenti su un database presente sul suo calcolatore e preferisce accedere all'aggiornamento degli impegni presi anche da remoto.

A che servono gli scenari?

A comprendere meglio il dominio di applicazione (principali aspetti, e aspetti secondari)

A trovare nuovi requisiti e definire nuove funzionalità ed in genere nuove applicazioni

A poter meglio specificare il processo di analisi del compito e quello di definire il modello del compito (relazioni fra compiti)

A valutare gli scopi al fine di soddisfare certi requisiti di usabilità e confrontare differenti soluzioni di progettazione

Un modello basato sui compiti (task) tipi di compiti

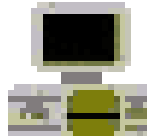
Compito dell'utente
(User Task)



- I compiti dell'utente (User Tasks)

Ciò che l'utente risolve nella veste di risolutore di problemi

Compito macchina
(ApplicationTask)



- Compiti dell'applicazione

Compiti completamente eseguiti dal sistema

Compito interazione
(Interaction Task)



- Compiti di interazione

Compiti che vengono eseguiti dall'interazione fra utente e sistemi secondo certo stile di interazione

Compito astratto
(Abstract Task)



Compiti astratti

Compiti che al momento della loro creazione non hanno una precisa attribuzione (chi sono gli esecutori?) e legame (in che modo?) con quali entità permetteranno la loro esecuzione

Specificazioni del ciclo di progettazione



Deve fornire un input per la costruzione del modello del compito

Qual è il risultato principale di un'analisi del compito?

Risposta: individuare i tipi di compiti (tasks) e il loro ruolo (cosa fanno)

Dagli esempi di scenari precedenti

Franco annota gli appuntamenti sulla sua agendina tascabile e non consulta spesso l'agendina. Franco dimentica spesso l'agenda sulla scrivania del suo studio, non vuole mancare agli impegni presi

I compiti individuati:

- consultare l'agenda in ogni luogo;
- aggiornare l'agenda in ogni luogo;

Uno strumento di ausilio all'identificazione del Task

Scenario

Scenario Description
scenariMefistoNuovo.txt

1 INTRODUCTION

This report describes three different scenarios for the Aerodrome Case Study selected in the MEFISTO PROJECT and described in Deliverable 1/3.

2 SCENARIOS

The scenarios refer to three different situations occurring in an airport. Two or more actors are involved in managing events critical from a safety point of view. In particular the following cases will be considered:

- * a runway incursion involving two departing aircraft
- * a runway incursion involving two arriving aircraft with a missed approach
- * a conflict on a taxiway involving two aircraft

2.1 SCENARIO 1 : RUNWAY INCURSION

Agents
Ground controller, tower controller, pilot aircraft 1, pilot aircraft 2

Rationale
The scenario is relevant in that it describes a failure in traffic management which leads to a safety

Roles specification

Add Delete

Role Name :

Departure Pilot

Ground
Tower
Departure Pilot
Departure co-ordinator
Arrivals co-ordinator
Arrival Pilot

Tasks Role specification

Add Delete Edit

Task Name :

pilot

- reach the holding position following the tax
- contact the tower controller
- wait for the clearance of take off from the t
- realises that something got wrong
- calls the ground controller
- pilot

Objects associated with selected Task

Add Delete Edit

Object Name :

plan of the airport, with runways and taxiways
signals on the airport area indicating runways and
a manual describing all the procedures which app

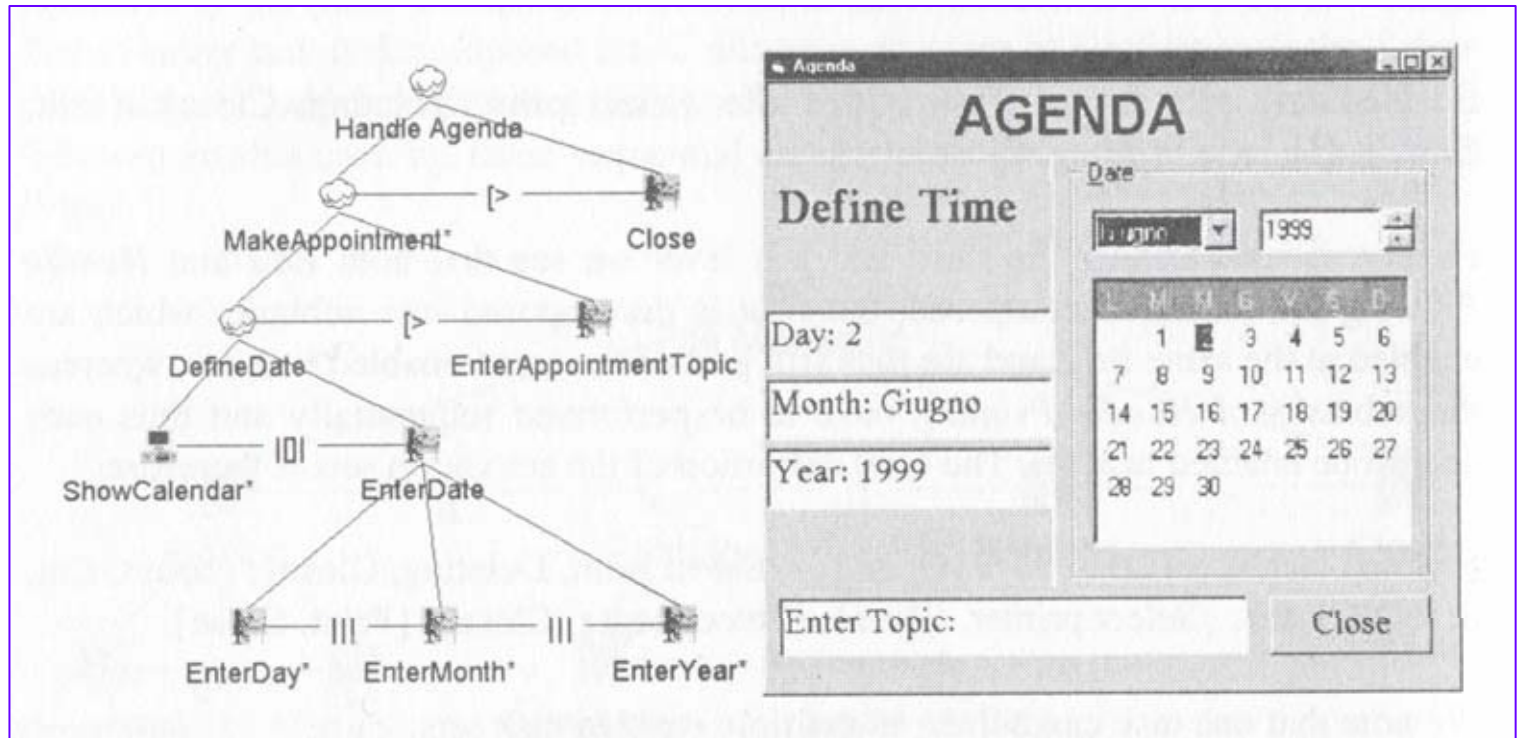
Role Objects List

plan of the airport, with runways and taxiways (pilot
signals on the airport area indicating runways and
a manual describing all the procedures which app

Questo report
descrive tre
differenti scenari
presenti in un
aeroporto

Due principali caratteristiche del modello del compito

- Rappresenta un processo di riduzione per specificazione: da i task più astratti ai task eseguibili (senza ulteriori specificazioni)
- Rappresenta le relazione temporale fra i task (astratti e non)



RELAZIONI TEMPORALI (I)

- $T1 \parallel T2$ Task indipendenti o concorrenti (es. visualizzare schermo e muovere il mouse);
- $T1 [] T2$ Scelta (si può scegliere o il task T1 o T2);
- $T1 [> T2$ Disabilitazione (il primo task T1 viene disattivato non appena la prima azione del secondo task T2 viene eseguita);
- $T1 \gg T2$ Sequenzialità (se non finisce il primo task T1 non può iniziare T2)

RELAZIONI TEMPORALI (II)

- $T1 \ll T2$ Sequenzialità con passaggio di informazione (se non termina $T1$ non può iniziare $T2$ - $T1$ passa delle informazioni a $T2$);
- $[T]$ Il task T è opzionale (il riempimento dei campi di una form
- T^* Iterazione (il task viene iterato)

UNA IMMEDIATA PROPRIETA': L'EREDITARIETA' DELLE RELAZIONI TEMPORALI

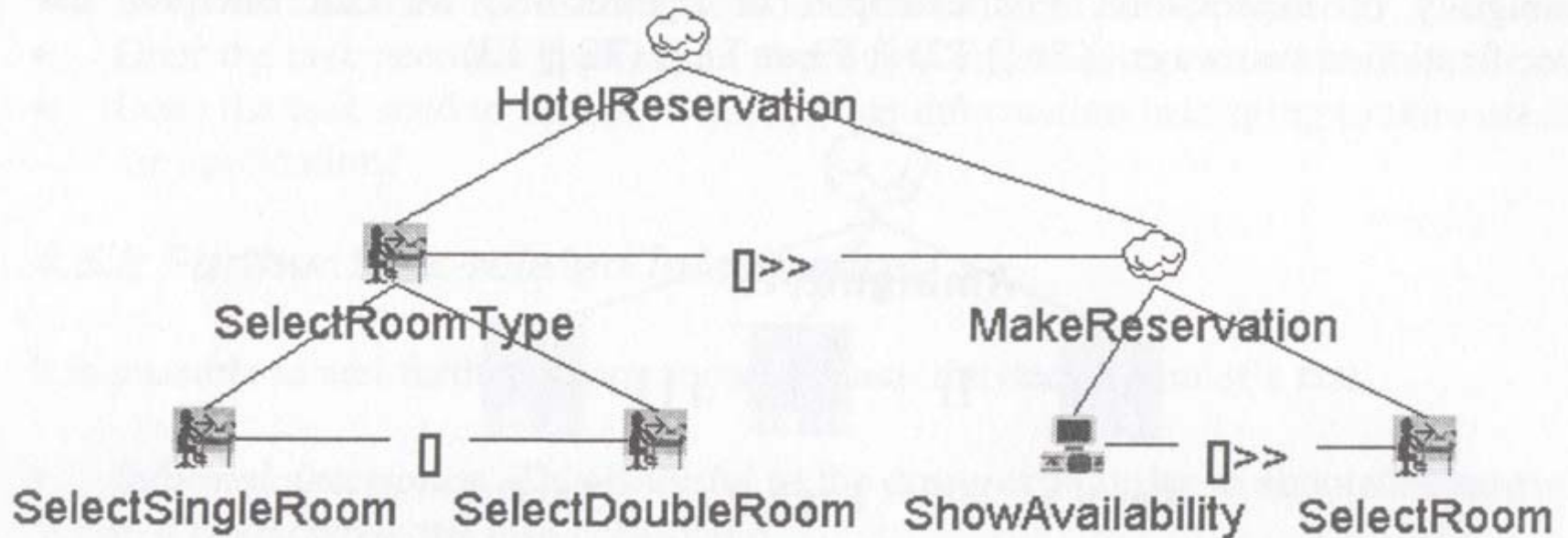
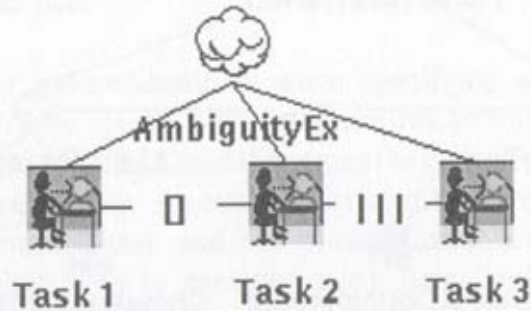


Figure 4.3: An example of inheritance of temporal relationships.

II PROBLEMA DELL'AMBIGUITA'



Due possibili interpretazioni

Figure 4.1: An example of ambiguity.

$(T1 [] T2) ||| T3$

To solve the ambiguity problem we have two options. The first option is to use the priority order among operators:

choice > parallel composition > disabling > enabling

oppure

$T1 [] (T2 ||| T3)$

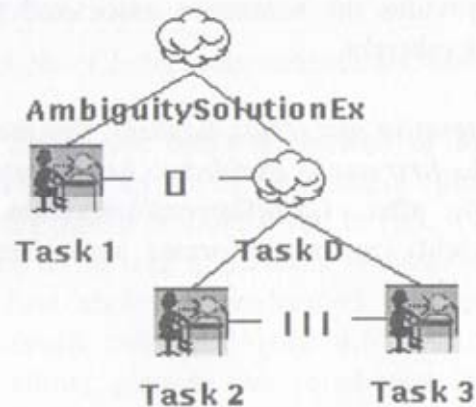


Figure 4.2: A solution to solve ambiguity.

Un'importante nozione per i sistemi interattivi:

i compiti disponibili (enable tasks)

RELAZIONI TEMPORALI FRA TASK E DISPONIBILITA' (I)

Il caso T1 ||| T2

I task T1 e T2 sono indipendenti. All'inizio sono entrambi disponibili. Quando uno dei due viene eseguito si rende disponibile solo l'altro. Quando entrambi vengono eseguiti il processo termina.

Quando viene adoperato?

Quando si vuole che ogni elemento di una serie di task vengano eseguiti. Esempio il riempimento di una form.

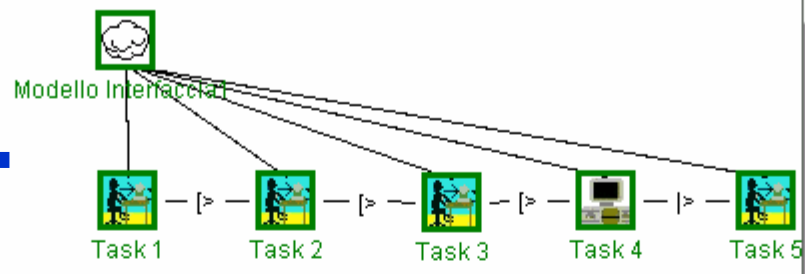
RELAZIONI TEMPORALI FRA TASK E DISPONIBILITA' (I)

Il caso T1 [$>$ T2 (il primo task T1 viene disattivato non appena la prima azione del secondo task T2 viene eseguita).

Il task T1 e T2 sono sempre disponibili

Quando viene adoperato?

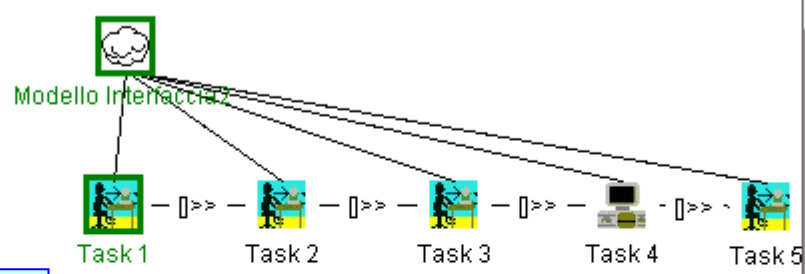
Ogni volta che si vuole fornire nell'interfaccia una possibilità di intervento (di controllo) all'utente per mettere fine ad un processo (o una serie di processi) senza che i processi disattivati forniscano informazione.



Overview

Enabled Tasks

- Enabled Tasks
- Task 1
- Task 2
- Task 3
- Task 4



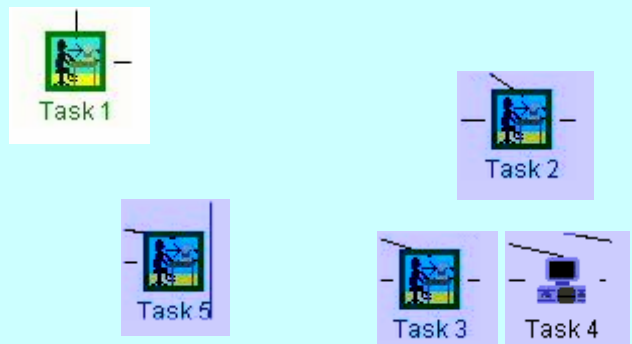
Overview

Enabled Tasks

- Enabled Tasks
- Task 1

Per entrambi - il passaggio di informazione implica una disabilitazione dei task a cui è destinata l'informazione

Interfaccia 2



Initial State

Commands

Back	Forward
Perform Task	Scenario Performed
Load Scenario	Stop Simulator



Compito astratto



**Compito macchina:
un programma**



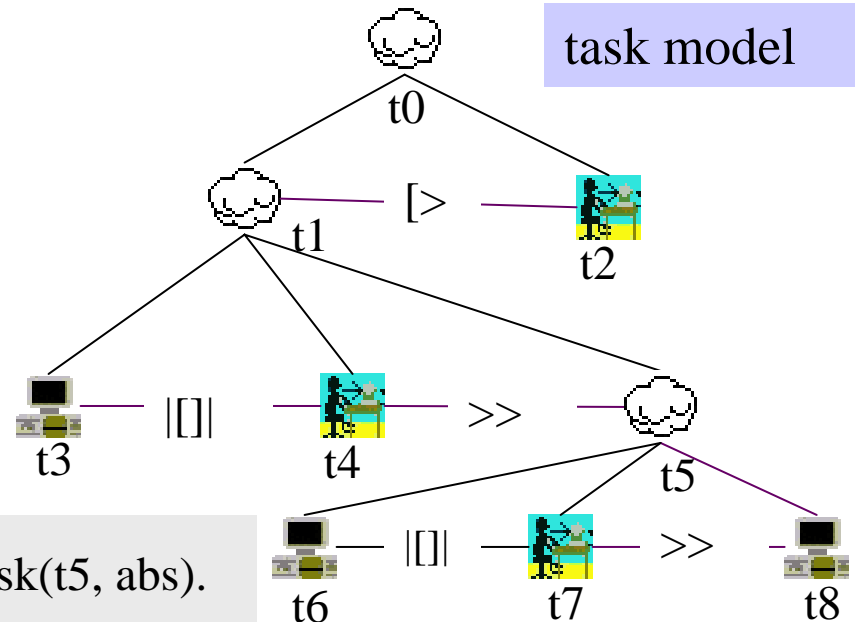
Compito dell'uomo



**Un'interazione uomo
macchina**

legenda

task model



$\text{task}(t_0, \text{abs}) \leftarrow \text{task}(t_1, \text{abs}) [>] \text{task}(t_2, \text{int}).$

$\text{task}(t_1, \text{abs}) \leftarrow \text{task}(t_3, \text{app}) || [] \text{task}(t_4, \text{int}) \gg \text{task}(t_5, \text{abs}).$

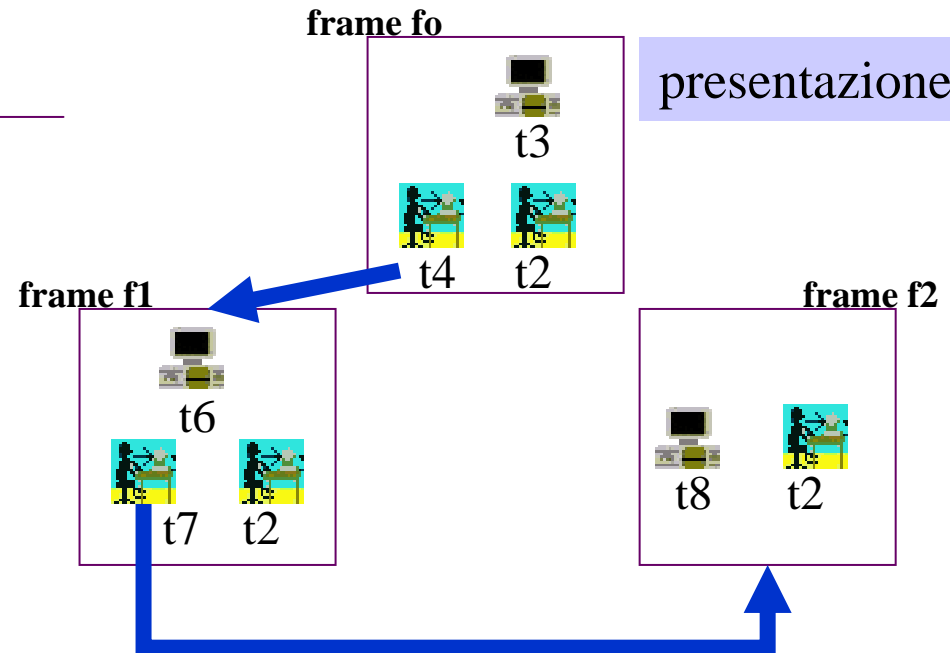
$\text{task}(t_5, \text{abs}) \leftarrow \text{task}(t_6, \text{app}) || [] \text{task}(t_7, \text{int}) \gg \text{task}(t_8, \text{app}).$

presentazione

$\text{frame}(f_0, [t_3, t_4, t_2]).$

$\text{frame}(f_1, [t_6, t_7, t_2]).$

$\text{frame}(f_2, [t_8, t_2]).$



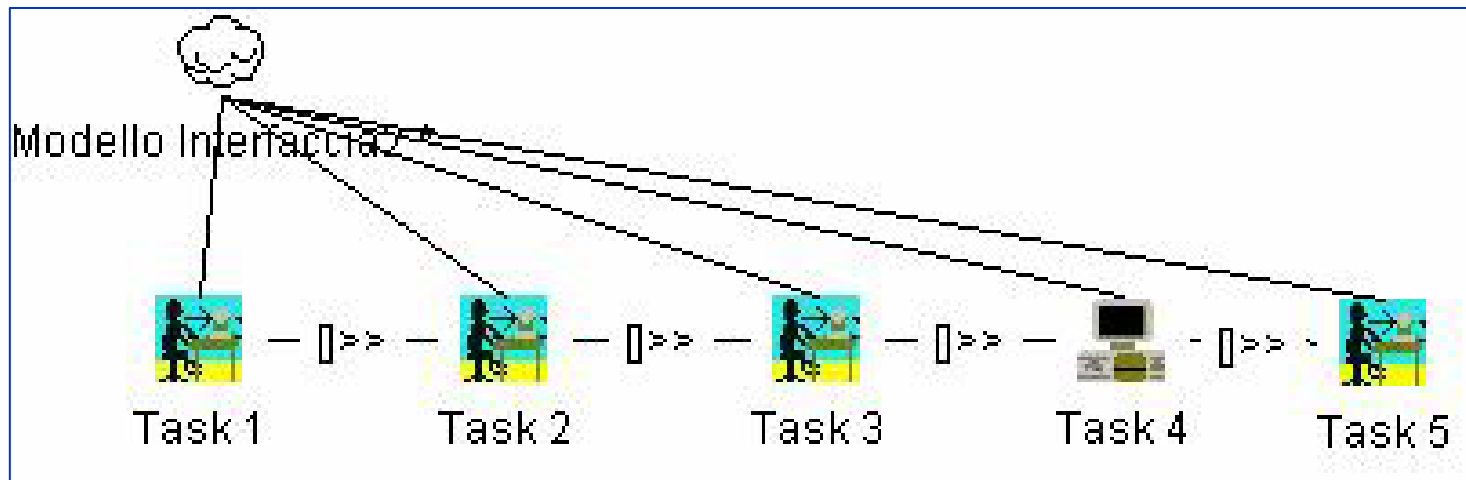
Task Iterativi

Due tipi:

Iterazione del nodo padre

Iterazione del nodo figlio

Iterazione del nodo padre

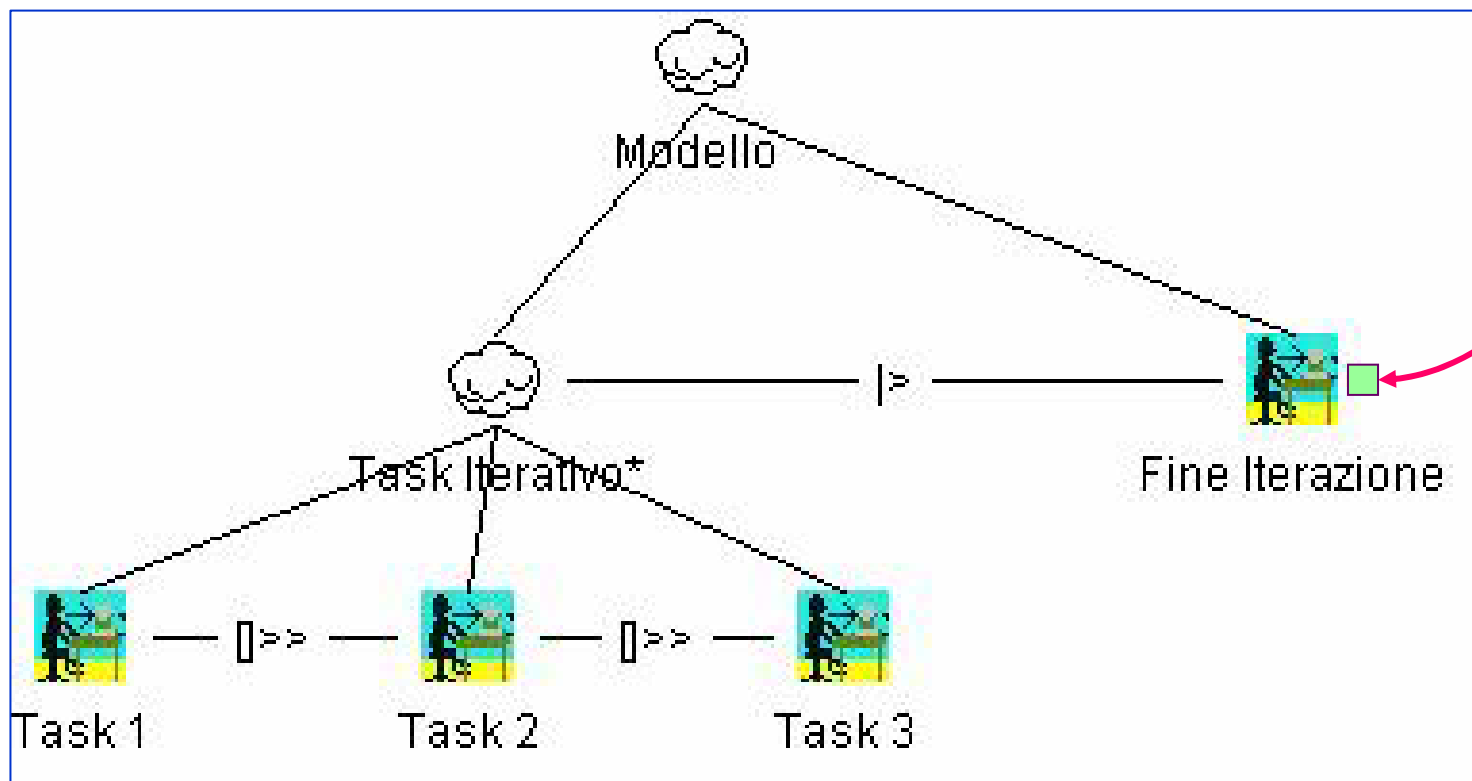


Dopo che viene eseguito il Task5 viene rieseguito il Task1

Uso del Task

Tutte le volte che si vuole proporre all'utente di ripetere una serie di operazioni.

- Occorre fornire sempre associare un task di controllo ad un task iterativo (padre)
- controllo ad un task iterativo (padre)

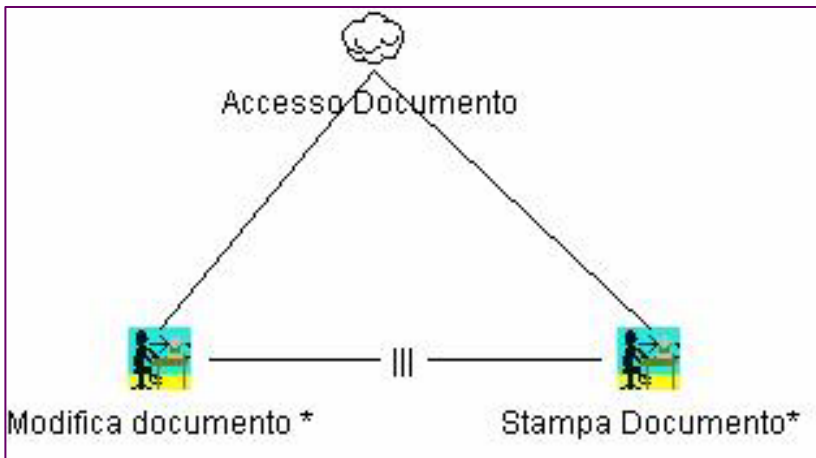


Iterazione del nodo figlio

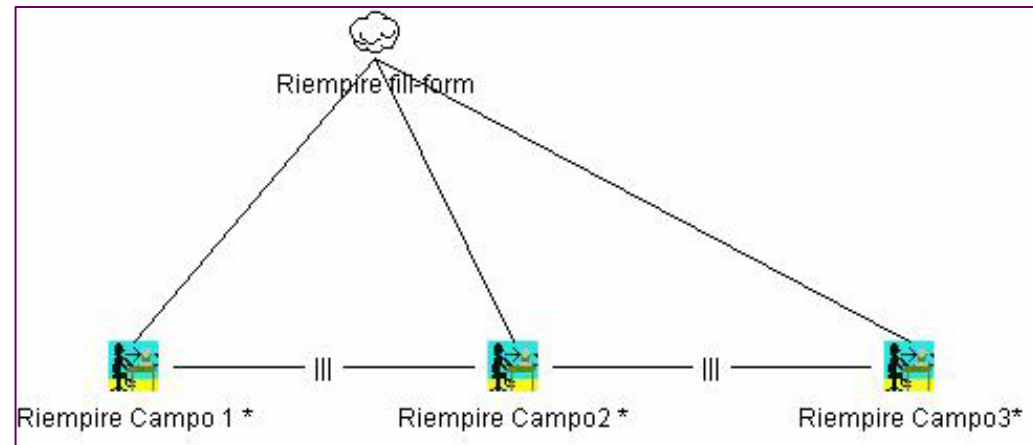
Quando viene usato?

Per avere sempre i task disponibili dopo una certa attivazione

Esempio A

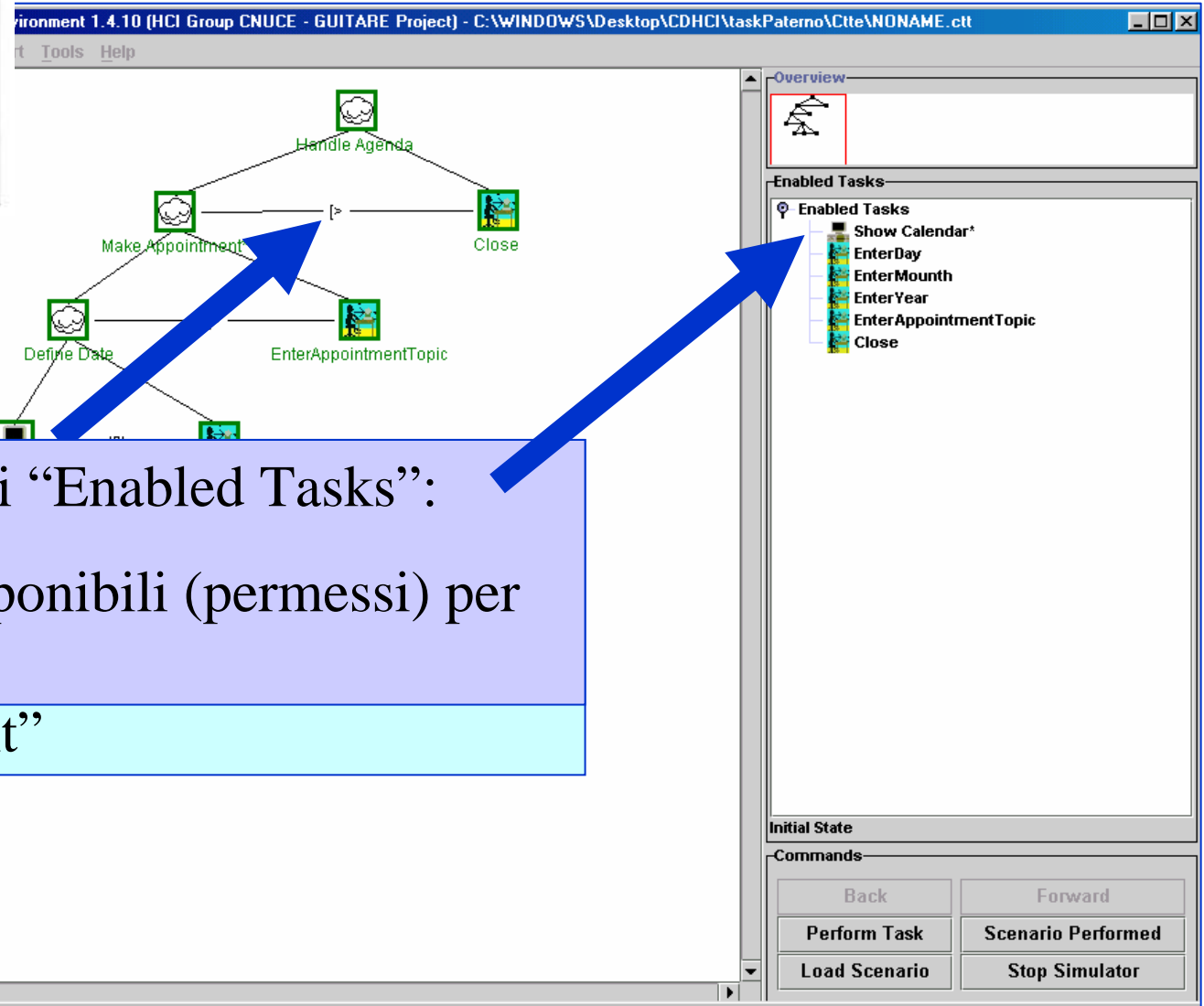
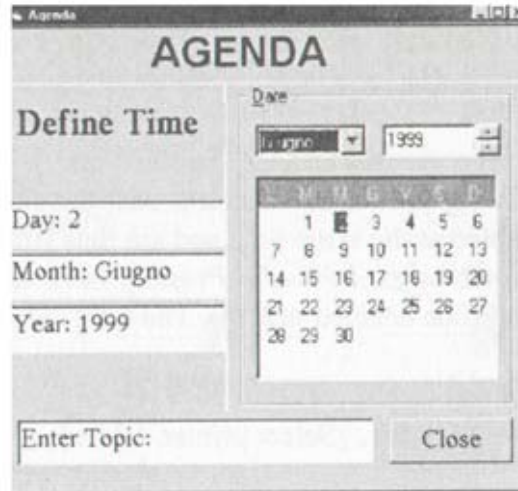


Esempio B



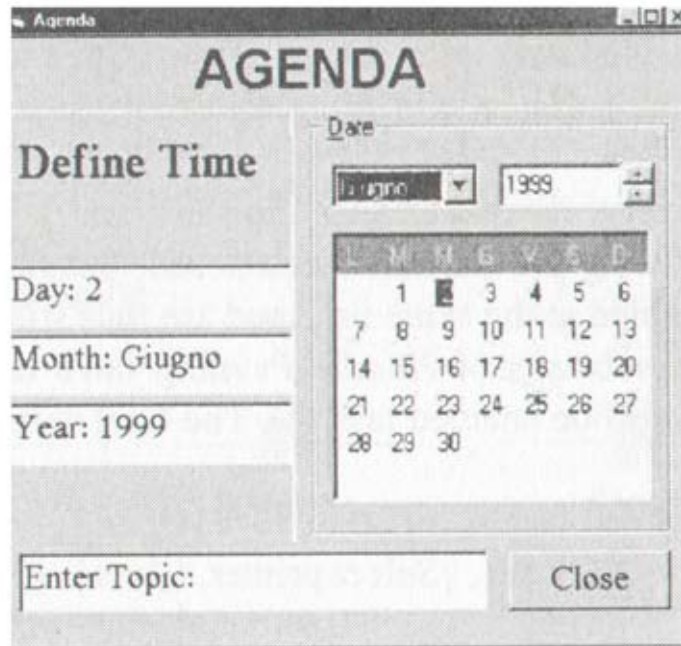
Ovviamente, occorre sempre applicare dei task di controllo

Ancora sui task disponibili



2- Interpretazione di “Enabled Tasks”:
i tasks che sono disponibili (permessi) per
l’esecuzione
“Make Appointment”

L'interfaccia "Agenda" ha un problema. Quale?

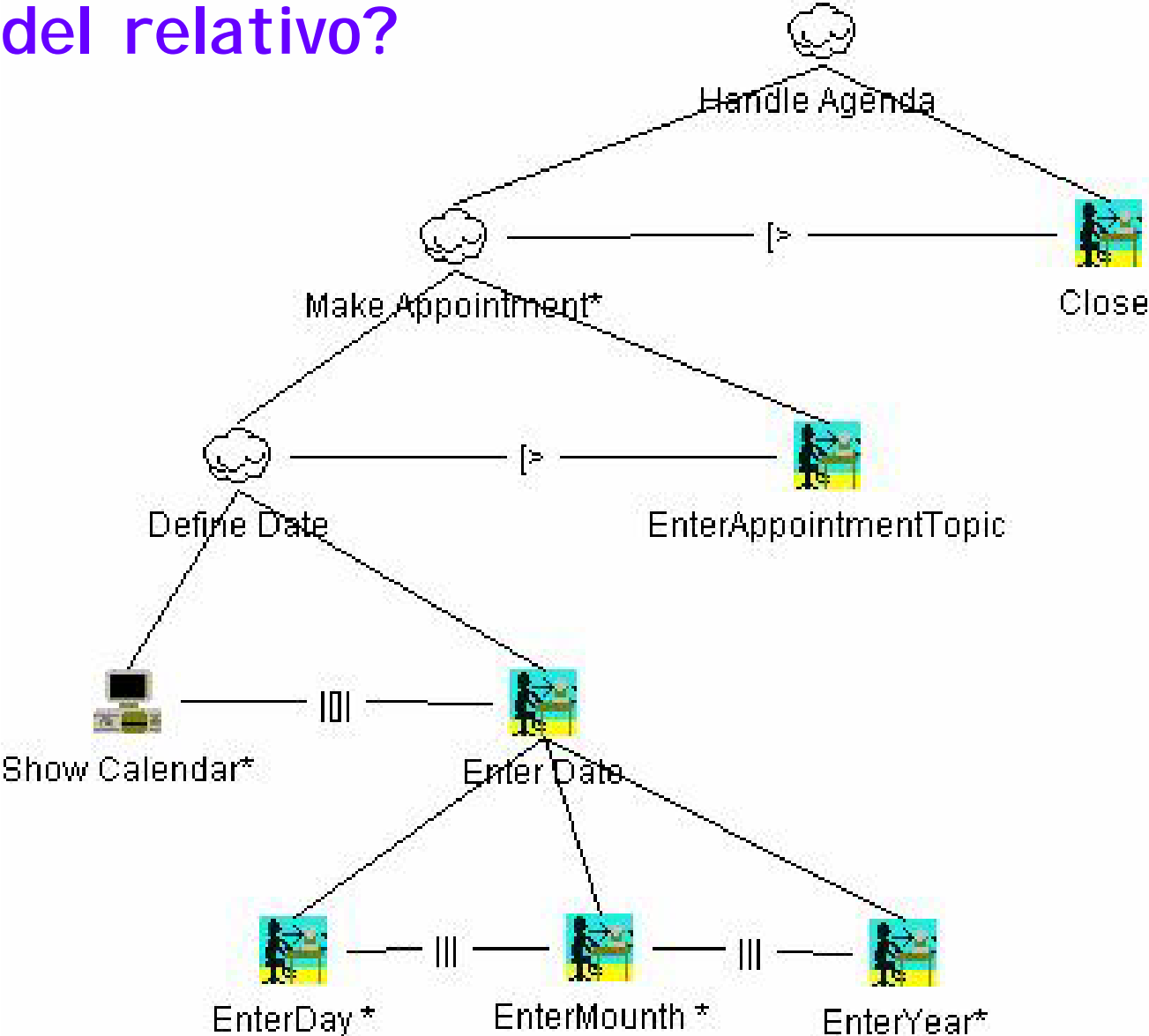


Si può premere il tasto
“Close” senza che
l’operazione di ingresso
dei dati sia completata

Soluzione?

Non a caso le interfacce vengono costruite in maniera tale che al posto di “Close” vi è sempre la coppia:
“Registra” - “Cancella”

Come può essere modificato il Task Model relativo?



Un normale processo di progettazione

Elicitazione



Creazione del Task Model



Creazione dell'interfaccia

Un esercizio obbligatorio

Creare il Task Model



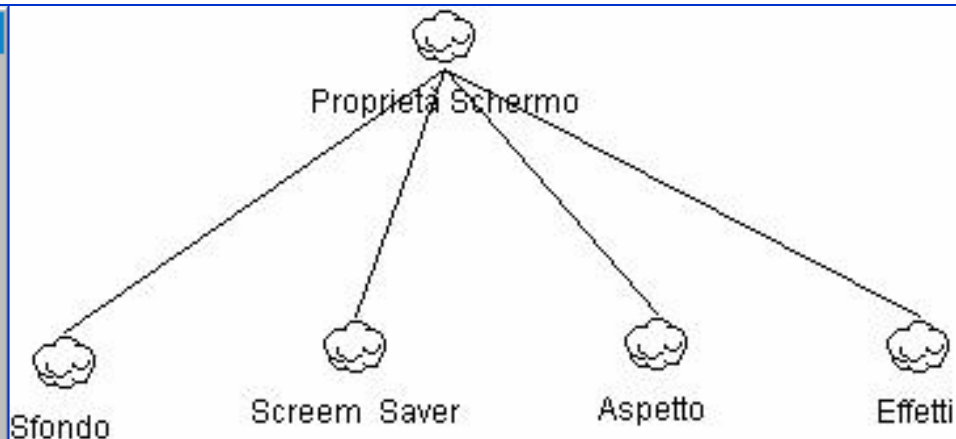
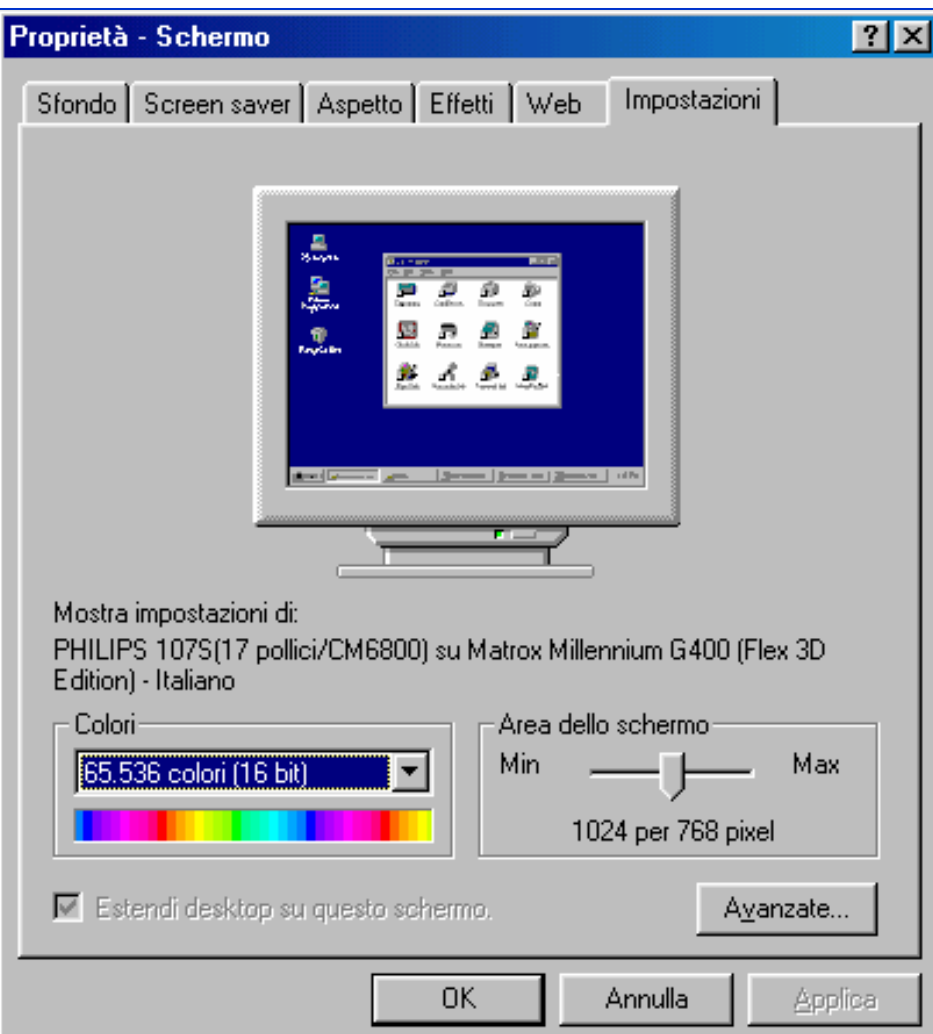
Interfaccia

L'esercizio

Interfaccia



Task Model



CTTE (ConcurTaskTree Environment) è un strumento per lo sviluppo di interfacce

Funzionalità

- Permette ai progettisti di inserire e cambiare il modello del task nella rappresentazione prevista dalla notazione;
- Possiede capacità grafiche per la costruzione e manipolazione del task model;
- Possiede alcune capacità per il controllo di consistenza;
- Permette il salvataggio dei modelli di task (o parte di essi) in vari formati;
- Fornisce la possibilità di eseguire una simulazione del task model,

File Edit View Info Insert **Tools** Help

Informal to formal description Ctrl-D
Start Task Model Simulator F4
Check Model Structure F7
Generate Enabled Task Sets Ctrl-G

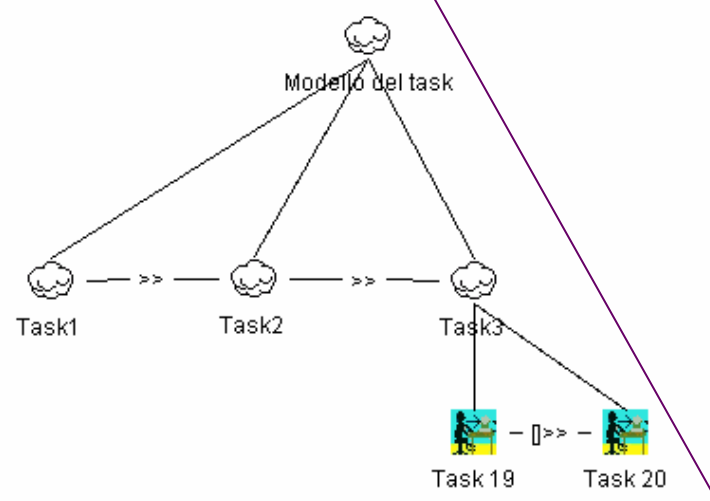
Global view Local view

10% 100% 200%

Dialog 12

Current Task Identifier: Task 19 Type: Frequency:

Overview



Check Consistence Results

Error in cooperative: The Task 'Task1 ' can not be "Abstract" because is a leaf in the tree
Error in cooperative: The Task 'Task2 ' can not be "Abstract" because is a leaf in the tree
2 error(s) found

Double click on the error line to locate the error.

Ok

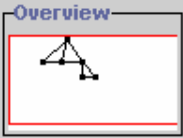
- File
 - Edit
 - View
 - Info
 - Insert
 - Tools
 - Help
- | | |
|---------------------|--------------|
| New | Ctrl-N |
| Open | Ctrl-O |
| Save | Ctrl-S |
| Save As | Ctrl+Shift-S |
| Save Subtree As | |
| Save CTT As XML | |
| Save Tree As JPG | Ctrl-J |
| Save Subtree As JPG | |
| Print | Ctrl-P |
| Quit | Ctrl-Q |

Global view Local view

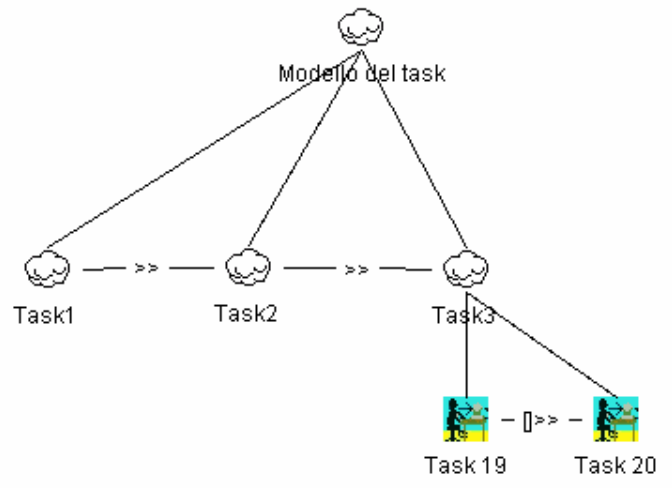
10% 100% 200%

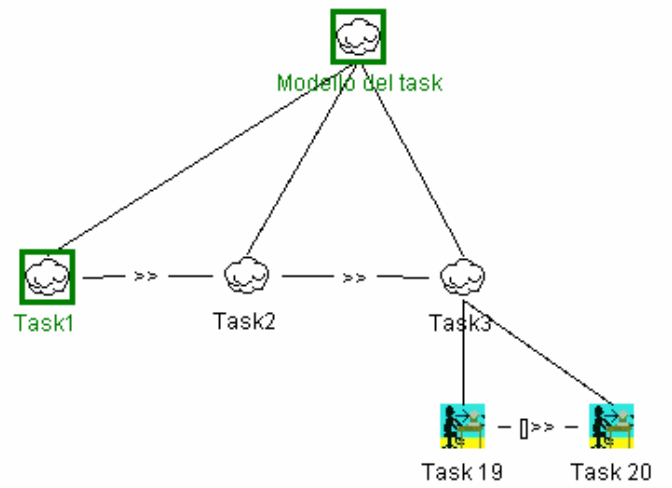
Dialog 12

Category: Interaction Type: Frequency:



- High Priority
- []
 - []
 - []
 - []
 - []
 - []
 - []
- Low Priority
- []
 - []
 - []
 - []





Check Consistence Results

Warning in cooperative: The Task 'Task1 ' can not be "Abstract" because is a leaf in the tree
Warning in cooperative: The Task 'Task2 ' can not be "Abstract" because is a leaf in the tree
Check Terminated with No Errors...
2 warning(s) found

Ok

Overview

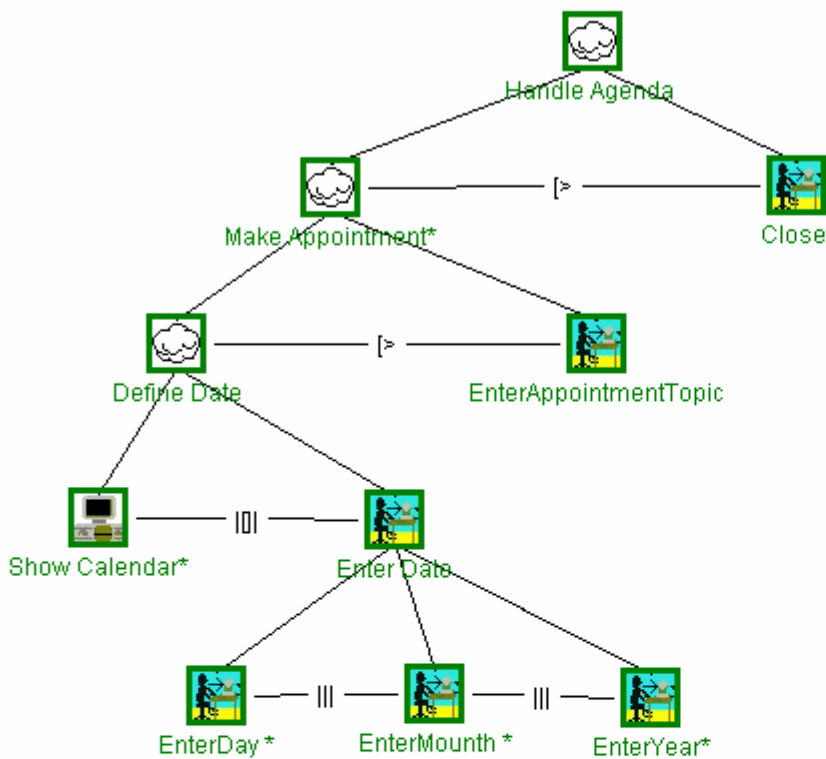
Enabled Tasks

- Enabled Tasks
 - Task1

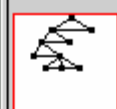
Initial State

Commands

Back	Forward
Perform Task	Scenario Performed
Load Scenario	Stop Simulator



Overview



Enabled Tasks

Enabled Tasks

- Show Calendar*
- EnterDay
- EnterMounth
- EnterYear
- EnterAppointmentTopic
- Close

Scenario Performed

Show Calendar*

EnterDay

EnterMounth

EnterYear

EnterAppointmentTopic

Save Cancel Close

Task Performed : EnterAppointmentTopic

Commands

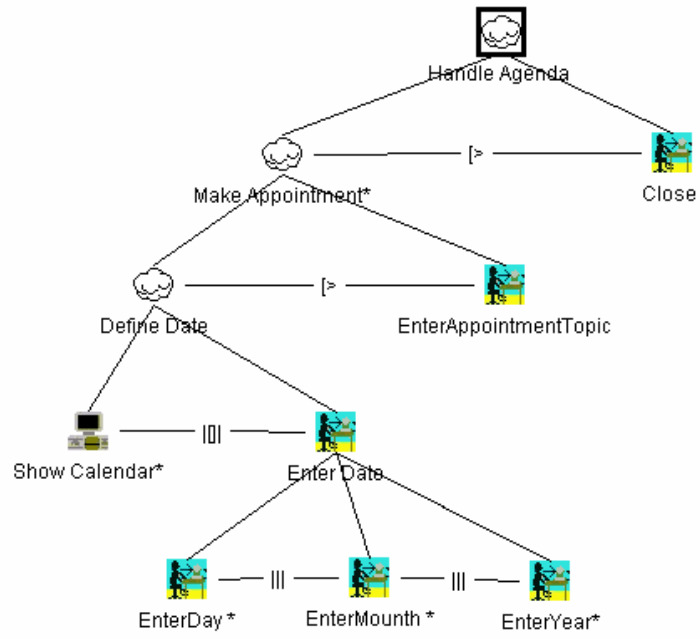
Back	Forward
Perform Task	Scenario Performed
Load Scenario	Stop Simulator



High Priority

Low Priority

Navigation icons: Home, Back, Forward, Search, etc.



List of Enabled Task Sets:

- Level 0: {Handle Agenda }
- Level 1: {Make Appointment, First(Close)}
 - {Body(Close)}
- Level 2: {Define Date, First(EnterAppointmentTopic), Close }
 - {Body(EnterAppointmentTopic), Close }
- Level 3: {Show Calendar*, Enter Date, EnterAppointmentTopic, Close }
- Level 4: {Show Calendar*, EnterDay , EnterMounth , EnterYear, EnterAppointmentTopic, Close }

```
- <ETSlist>
- <Level number="0">
  - <ETS>
    <task name="Handle Agenda" type="" />
  </ETS>
</Level>
- <Level number="1">
  - <ETS>
    <task name="Make Appointment" type="" />
    <task name="Close" type="First" />
  </ETS>
- <ETS>
  <task name="Close" type="Body" />
</ETS>
</Level>
- <Level number="2">
  - <ETS>
    <task name="Define Date" type="" />
    <task name="EnterAppointmentTopic" type="First" />
    <task name="Close" type="" />
  </ETS>
- <ETS>
  <task name="EnterAppointmentTopic" type="Body" />
  <task name="Close" type="" />
</ETS>
</ETS>
</Level>
- <Level number="3">
```

```
- <Level number="3">
  - <ETS>
    <task name="Show Calendar*" type="" />
    <task name="Enter Date" type="" />
    <task name="EnterAppointmentTopic" type="" />
    <task name="Close" type="" />
  </ETS>
</Level>
- <Level number="4">
  - <ETS>
    <task name="Show Calendar*" type="" />
    <task name="EnterDay" type="" />
    <task name="EnterMounth" type="" />
    <task name="EnterYear" type="" />
    <task name="EnterAppointmentTopic" type="" />
    <task name="Close" type="" />
  </ETS>
</Level>
</ETSlist>
```