

1 Manuale utente

Dopo aver installato la cartella “progetto” nel file system per creare una simulazione di un cliente cliccare due volte sul file jar eseguibile “GUIxml”, apparirà un’interfaccia utente come quella mostrata nella figura sottostante :

XML GENERATOR

Proprietà del Cliente

Nome

Conto

Cassa

Casa

Inserisci

Transazioni del Cliente

Bancomat

Prelievo

Versa in Conto

Versa in Cassa

Inserisci

Aggiungi Cliente **Aggiungi**

Nuova simulazione **Cancella**

Crea il File xml per la simulazione del Cliente

Nome del File .xml

Genera

Aggiungi clienti ad un file xml esistente

File

Apri **Modifica**

La parte superiore dell'interfaccia è destinata alla creazione di un nuovo cliente : è possibile inserire una serie di dati che riguardano il nome e le proprietà che questi possiede e nella parte affianco tutti i servizi che si vogliono far simulare al cliente, nell'ordine preferito dall'utente.

Una volta terminato l'inserimento di tali dati è possibile aggiungere il cliente alla lista e proseguire con la creazione di un altro cliente.

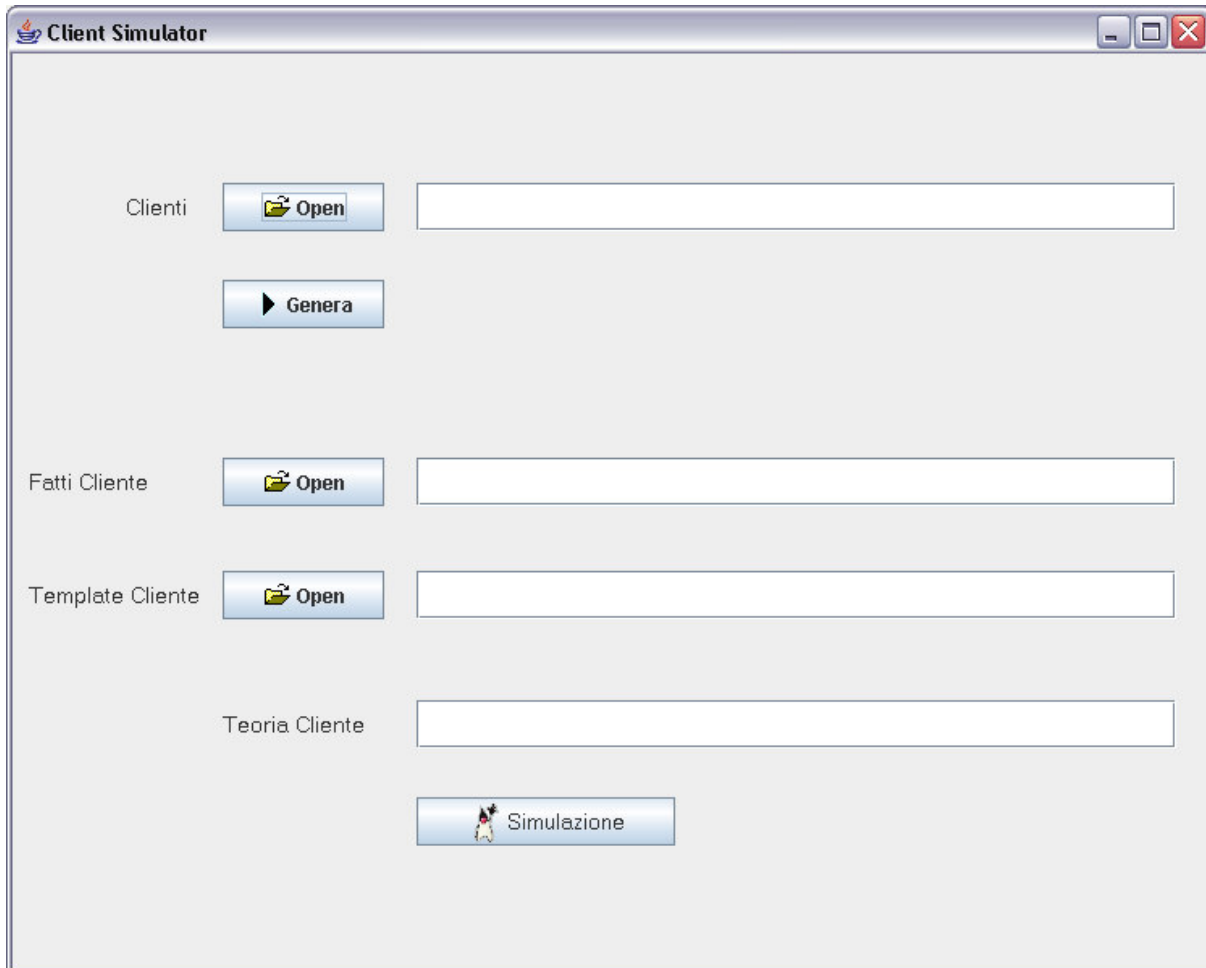
Una volta inseriti i clienti può essere generato il file xml che conterrà i loro dati e servizi personali con un semplice click sul tasto "Genera".

Nel caso in cui si disponga già di un file xml creato in precedenza si può riaprire per aggiungere un nuovo cliente con la modalità vista sopra.

A questo punto si può passare alla creazione delle teorie logiche di fatti prolog per ogni cliente da simulare.

Per far ciò occorre aprire una seconda interfaccia con un doppio click sul file jar eseguibile "Parser".

Verrà aperta una schermata come quella che segue in figura sottostante:

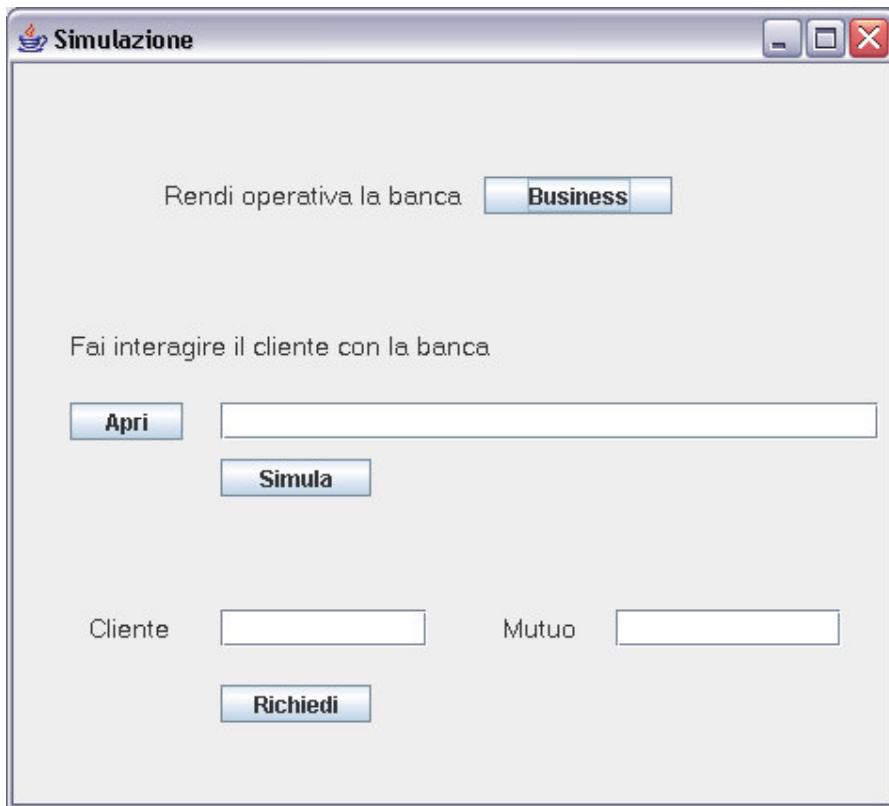


Per prima cosa occorre aprire il file xml generato o modificato in precedenza e con un click sul pulsante “Genera” verranno automaticamente generate le teorie logiche prolog di soli fatti che corrispondono ai dati, alle proprietà ed ai servizi di cui il cliente sarà costituito in fase di simulazione.

Per creare il file finale che rappresenta il cliente nella sua interezza occorre associare i fatti del cliente e quindi le sue azioni al modello del cliente (file “Template” all’interno della cartella progetto).

Una volta che sono stati selezionati i due file teoria e regole occorre associarli in un unico file che rappresenta il cliente, quindi premere sul pulsante “Simulazione”.

Ora si è pronti per far interagire il cliente con la banca, ma prima occorre rendere operativa quest’ultima per cui aprire l’interfaccia di simulazione con un doppio click sul file jar eseguibile “Simulazione” apparirà una schermata come quella che segue:



Nella prima parte è richiesto un semplice click per dar vita al sistema bancario a questo punto il sistema è operativo.

Nella parte sottostante occorre selezionare il file relativo al cliente da simulare e premendo il tasto “simula” la simulazione avrà inizio.

Per osservare il comportamento dell'interazione del cliente con la banca occorre far partire l'infrastruttura "TucSoN" e due "Inspector" uno per il centro di tuple della banca e uno relativo alle proprietà del cliente, denominati rispettivamente "banca" e "nomecliente".

Come si potrà notare i servizi che il cliente richiede verranno effettuati riportando alcuni risultati che rimangono nelle centro di tuple della banca è questa la storia del cliente.

Per cui dall'interfaccia di simulazione si può richiedere al cliente di simulare la richiesta di un mutuo bancario, quindi dall'interazione del cliente con la banca che ne verificherà le proprietà e la storia passata, emergerà un esito positivo o negativo per quel che riguarda l'esecuzione del servizio.

2 Esempi di simulazioni

La prima simulazione simula un cliente abbastanza disciplinato nei prelievi e con casa di proprietà, in questo caso si proverà, alla fine della simulazione, a richiedere il servizio di prestito pari ad un importo di 50000(\$) e questo dovrebbe dare esito positivo in quanto il cliente ha una proprietà ipotecabile che copre il valore richiesto e non ha prosciugato più di una volta il proprio contocorrente e deposito.

The screenshot shows a window titled "XML GENERATOR" with a standard Windows-style title bar. The window is divided into several sections for data entry and simulation control.

Proprietà del Cliente

Nome	<input type="text" value="raffaele"/>
Conto	<input type="text" value="1000"/>
Cassa	<input type="text" value="1000"/>
Casa	<input type="text" value="100000"/>

Transazioni del Cliente

Bancomat	<input type="text" value="1000"/>
Prelievo	<input type="text" value="1000"/>
Versa in Conto	<input type="text"/>
Versa in Cassa	<input type="text"/>

Aggiungi Cliente

Nuova simulazione

Crea il File xml per la simulazione del Cliente

Nome del File .xml

Aggiungi clienti ad un file xml esistente

Il file xml prodotto al termine di questa operazione è il seguente :

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <raffaele>
    <registration>raffaele</registration>
    <assets>
      <contocorrente>1000</contocorrente>
      <cassa>1000</cassa>
      <casa>100000</casa>
    </assets>
    <services>
      <bancomat>500</bancomat>
      <prelievo>500</prelievo>
    </services>
    <services>
      <versamentoConto>500</versamentoConto>
      <versamentoCassa>500</versamentoCassa>
    </services>
    <services>
      <bancomat>1000</bancomat>
      <prelievo>1000</prelievo>
    </services>
  </raffaele>
</root>
```

Tale file contiene tutti i dati del cliente che devono essere trasformati in termini di clausole di Horn del prim'ordine e compatibili con il sistema realizzato.

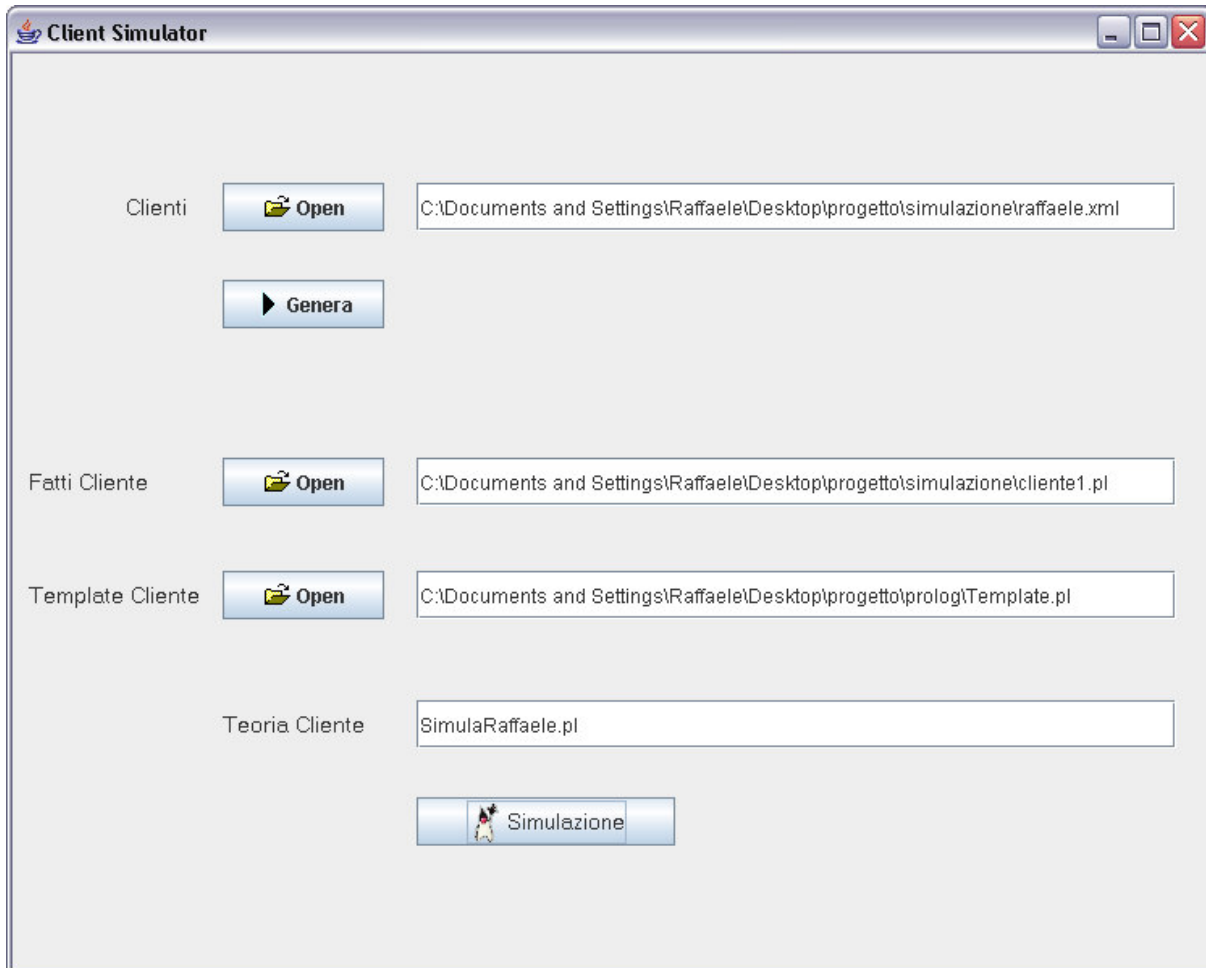
Quindi una volta selezionato tale file viene realizzato automaticamente un file chiamato "cliente1.pl" che definisce i dati passati dall'utente sottoforma di fatti logici prolog.

Il file è il seguente :

```
-----
simula:-
registration(raffaele),
conto(raffaele,1000),
cash(raffaele,1000),
home(100000),
bancomat(raffaele,500),
prelievo(raffaele,500),
versamento(raffaele,conto,500),
versamento(raffaele,cassa,500),
bancomat(raffaele,1000),
prelievo(raffaele,1000),
true. %EndOfFile
-----
```

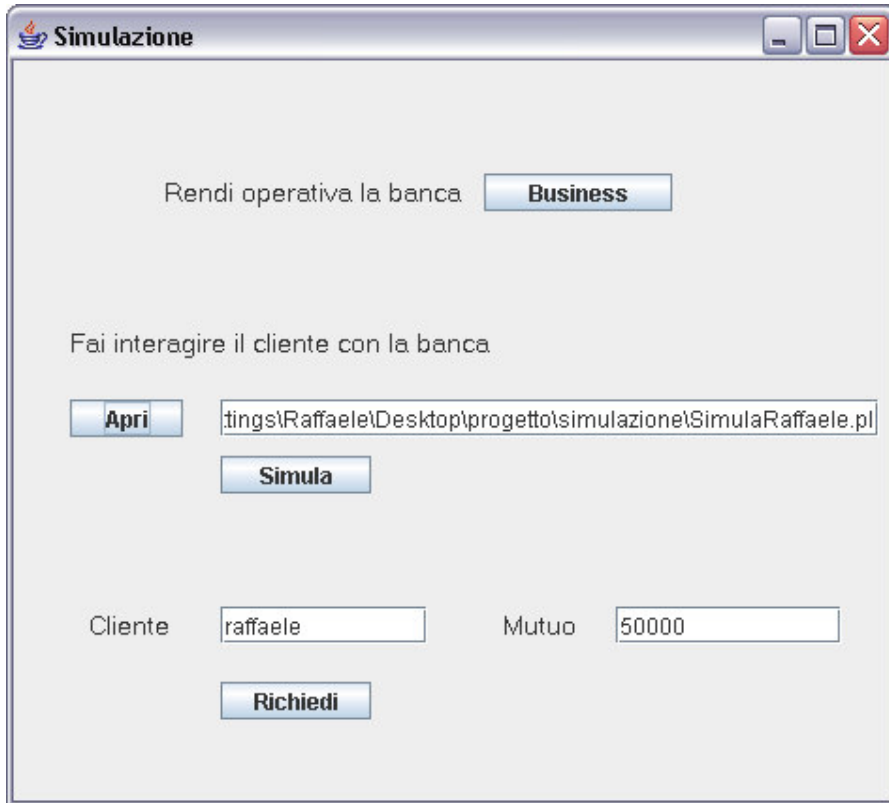
A questo punto occorre solo associare a tali fatti le regole ovvero il modello del cliente.

Tutto questo è mostrato nella figura sottostante:

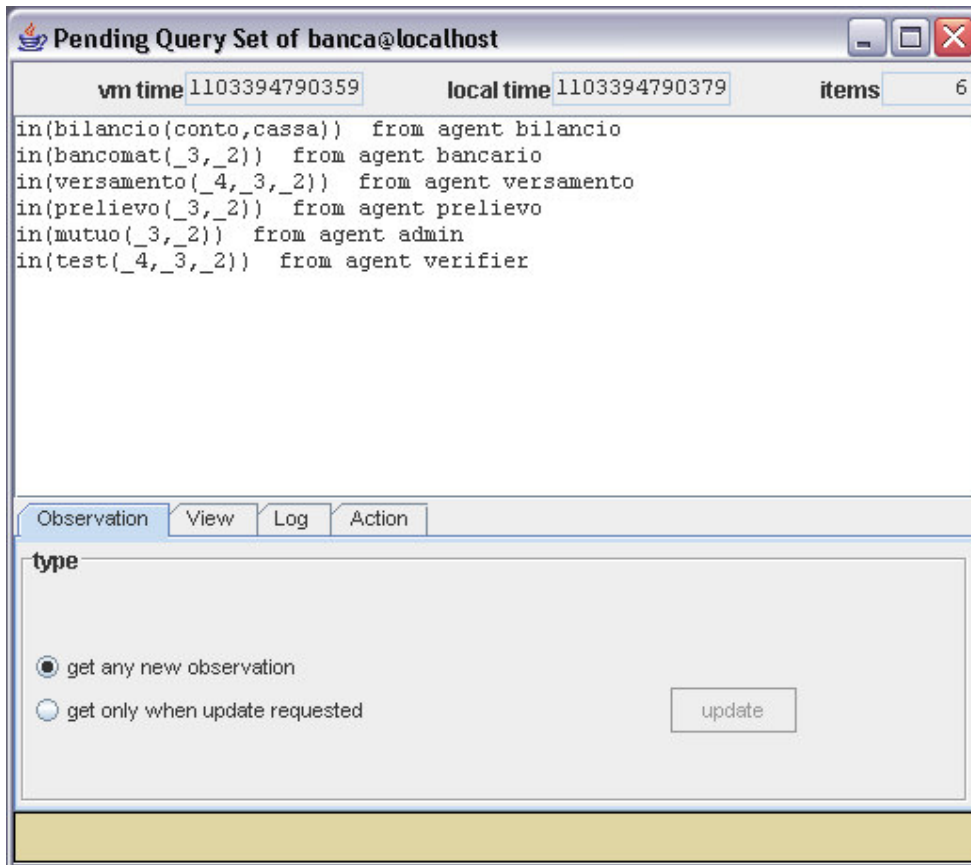


Il risultato di tale operazione e il file "SimulaRaffaele.pl" che consiste nel cliente vero e proprio da simulare.

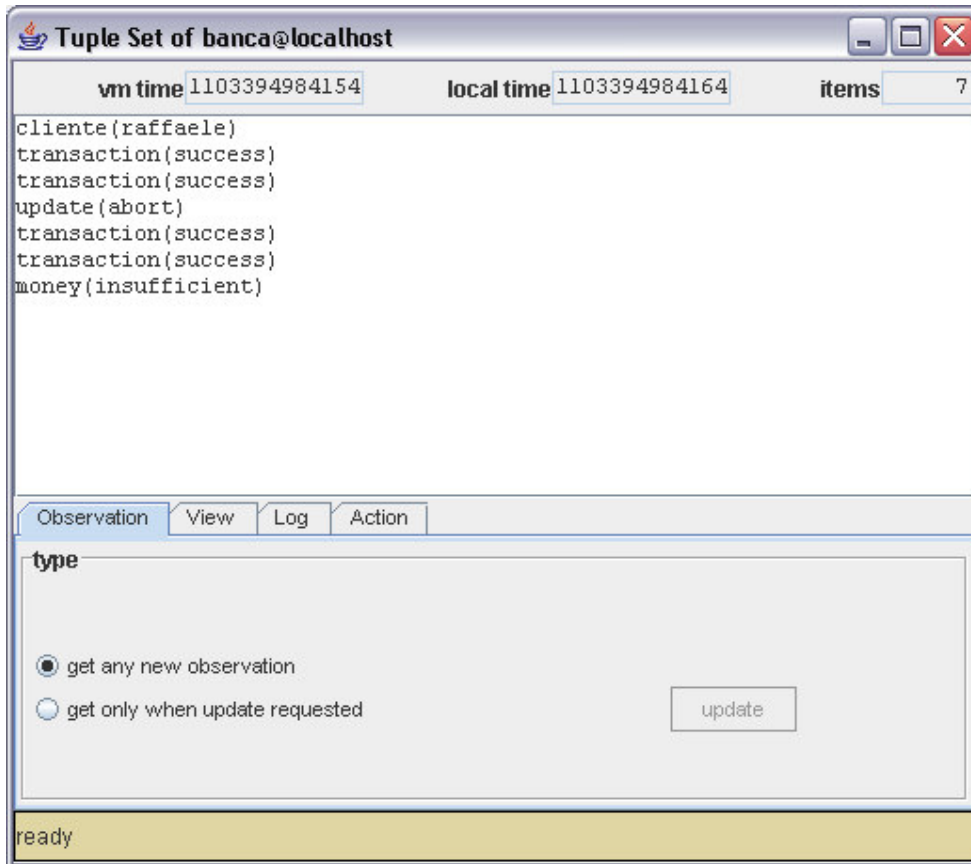
A questo punto non resta che dar vita al sistema bancario e simulare il cliente appena creato:



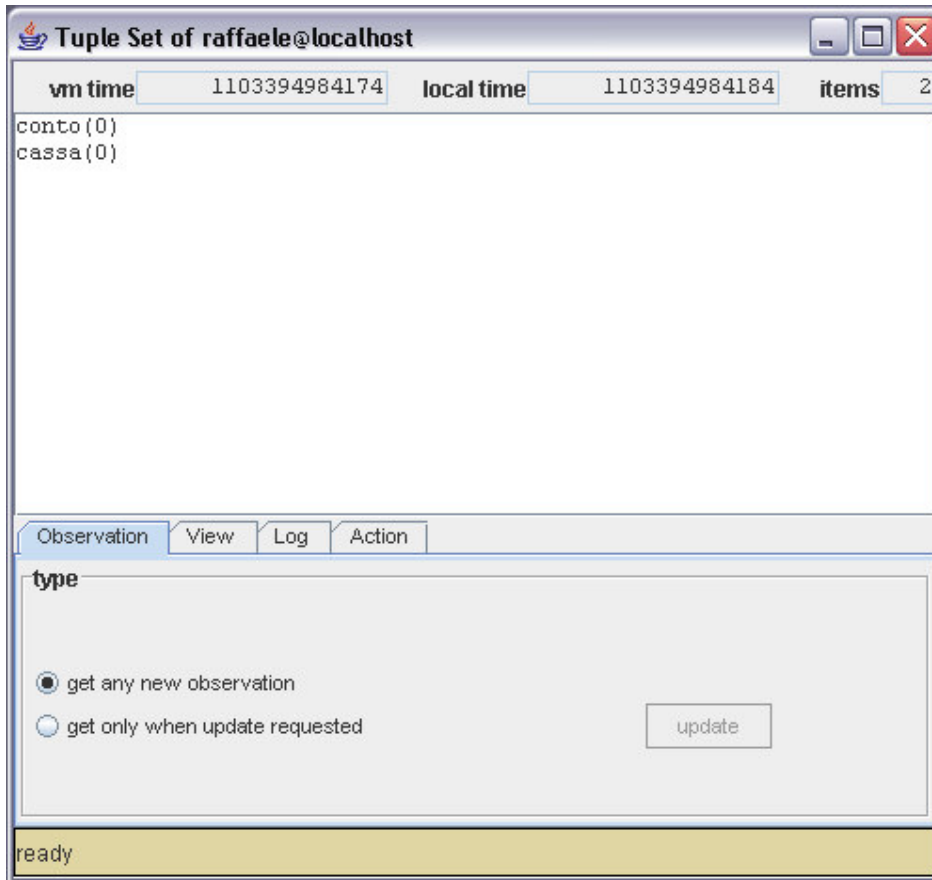
Dopo aver cliccato sul pulsante “business” abbiamo dato il via ai servizi bancari e gli agenti bancari che sono presenti a run-time sono mostrati in figura :



Prima della richiesta del mutuo abbiamo la seguente storia del cliente in banca :

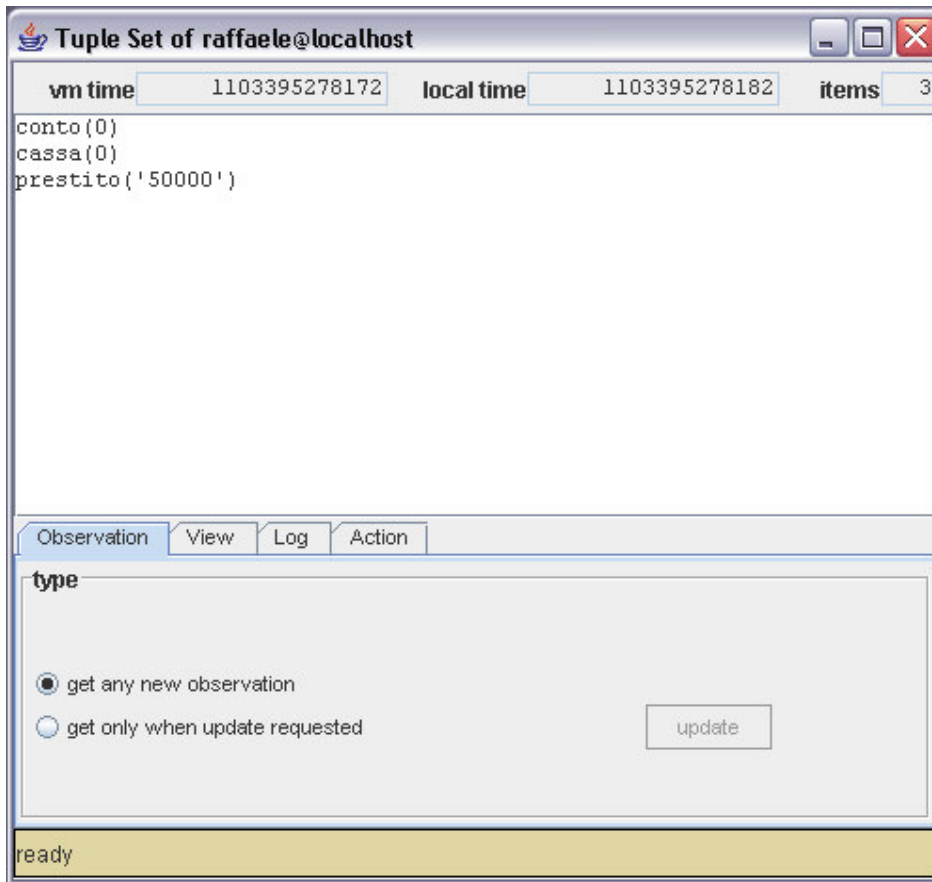


Per le proprietà abbiamo :



Ricordiamo che il cliente in questione aveva effettuato prima due prelievi e poi due versamenti della stessa cifra per poi eseguire due nuovi prelievi che annullavano l'importo complessivo.

Ora provando a chiedere il servizio di mutuo abbiamo il seguente risultato :



Viene effettuato il servizio con successo in quanto il cliente disponeva di un valore immobile opportuno e godeva della fiducia del direttore non avendo avuto un bilancio troppo negativo.

Eseguendo la stessa simulazione, con un cliente avente le stesse proprietà, ma con una storia finanziaria più disastrosa che porta alla sfiducia della banca nei suoi confronti abbiamo il rifiuto del servizio come mostrato nella figura sottostante in cui si evidenzia la storia finanziaria disastrosa prima della richiesta del mutuo :

The screenshot shows a window titled "Tuple Set of banca@localhost" with a status bar at the bottom indicating "ready". The main area displays a list of events:

vm time	local time	items
1103395809440	1103395809450	15

The events listed are:

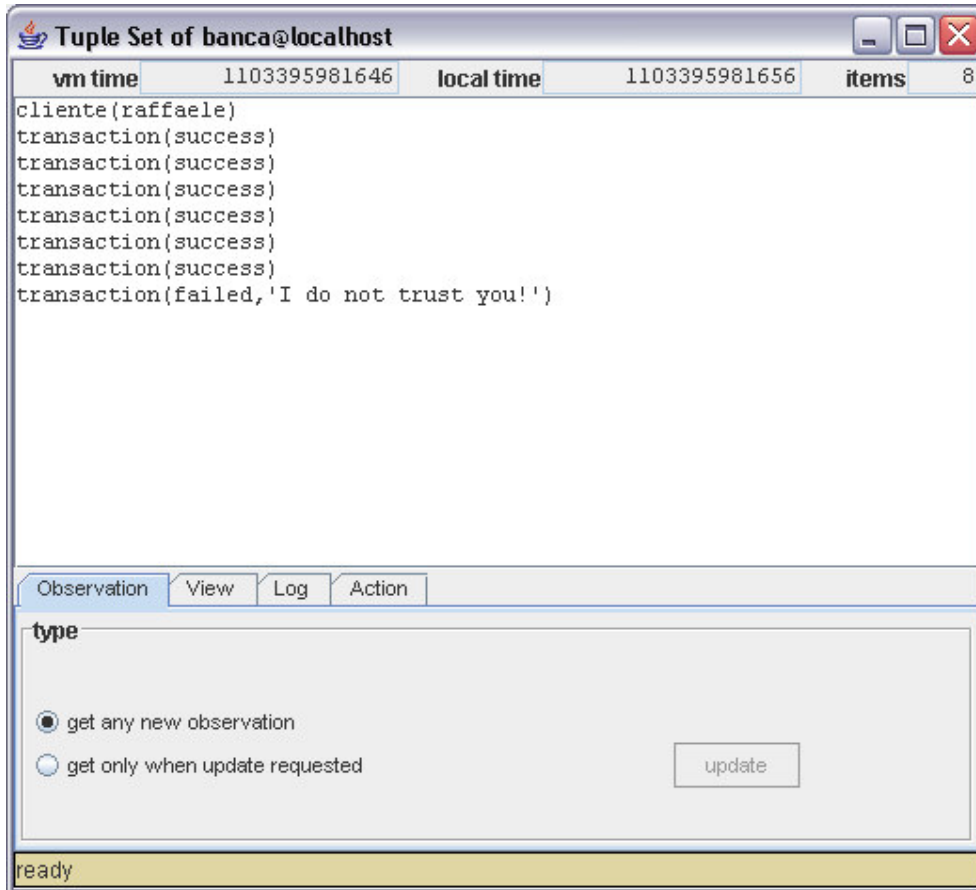
- cliente(raffaele)
- transaction(success)
- update(abort)
- money(insufficient)
- update(abort)
- transaction(success)
- transaction(success)
- money(insufficient)
- transaction(success)
- update(abort)
- money(insufficient)
- transaction(success)
- transaction(success)
- update(abort)
- money(insufficient)

Below the list, there are tabs for "Observation", "View", "Log", and "Action". Under the "Observation" tab, there is a "type" section with two radio buttons:

- get any new observation
- get only when update requested

An "update" button is located to the right of the radio buttons.

La risposta della banca alla richiesta di un prestito:



Nonostante il cliente abbia una proprietà che copre largamente il prestito richiesto, questo gli viene rifiutato per mancanza di fiducia.