

Laurea Specialistica in Informatica  
a.a. 2006-2007

Interazione Uomo-Macchina II:

**Interfacce Intelligenti**

Fiorella de Rosis

1

Introduzione

*Prima parte: Formalizzazione e Ragionamento*

- 1.1. Ragionamento logico:
  - Formalizzazione
  - Risoluzione
- 1.2. Ragionamento incerto
  - Reti Causali Probabilistiche
  - Reti dinamiche
  - Apprendimento di Reti

*Seconda parte: Modelli di Utente*

- 2.1. Modelli logici
- 2.2. Modelli con incertezza

*Terza parte: Interazione in linguaggio naturale*

- 3.1. Generazione di messaggi
  - Introduzione
  - Teorie
  - Metodi
- 3.2. Comprensione di messaggi

*Quarta parte: Simulazione di dialoghi*

Partiamo dall' esempio  
del trascritto di un dialogo telefonico  
(speech-based)  
sulla prenotazione di voli,  
che avevamo introdotto all'inizio del Corso.

Ragioniamo su come questo tipo di dialoghi possa  
essere simulato

3

S: A at American Express may I help you?  
U: yeah this is B. I need to make some travel arrangements  
[recording notification, charge number, travel advance, phone]  
S: ok and what do you need to do?  
U: ok on June sixth from San Jose to Denver, United  
S: leaving at what time?  
U: I believe there's one leaving at eleven o'clock in the morning  
S: yeah that's United flight four seventy  
U: ok  
S: leaves at eleven a.m. and arrives Denver at two twenty p.m. out of San Jose  
U: that's the one  
S: and a return?  
U: return June seventh  
S: leaving at what time?  
U: uh I believe there's one at five sixteen  
S: yeah that's United flight six fifty nine it arrives back into San Jose at six forty two  
U: ok  
S: let me just see what sort of fare I can get you on this  
U: all right  
S: ok the best fare I can get you on this would be five sixty eight round trip. That does  
have a twenty five percent change cancellation penalty on it  
U: ok  
S: And we'd have to ticket it by Tuesday  
U: right, that's fine  
[hotel, car, seating, mileage plus number] (Trascritto di un dialogo  
telefonico American  
Express)  
U: ok thank you very much  
S: thanks a lot

## Simulatori di Dialoghi vs Generatori di Monologhi

Generazione di Monologhi	Simulazione di Dialoghi
Il <i>modello dell'utente</i> viene definito inizialmente e non varia durante il processo di generazione	Il modello è funzione del contesto: scambi avvenuti e ragionamenti effettuati da S durante il dialogo
Gli <i>obiettivi comunicativi</i> vengono definiti inizialmente e quindi tradotti in un piano.	S ha degli obiettivi iniziali, ma può rivederli sulla base di nuove esigenze emerse nel corso del dialogo
Il <i>piano del discorso</i> viene definito inizialmente e quindi tradotto in un messaggio.	S definisce un 'piano globale' ma adatta al contesto i piani realizzati in ogni passo del dialogo.
Le <i>fonti d'informazione</i> sono: <ul style="list-style-type: none"> <li>• i piani o gli operatori di piano,</li> <li>• gli schemi per la generazione superficiale</li> <li>• il modello di U.</li> </ul>	Alle fonti necessarie per generare monologhi si aggiunge una 'storia del dialogo'.
Le <i>componenti principali</i> sono: un pianificatore, un (eventuale) revisore del piano e un generatore superficiale.	Alle componenti necessarie per generare un monologo si aggiunge un revisore del modello di utente, dei propri goal e belief e dei piani da realizzare <sup>5</sup>

## Complessità dei Dialoghi

I dialoghi possono essere più o meno semplici, a seconda dell' *iniziativa lasciata all'utente* (far domande o semplicemente rispondere alle domande del sistema) e della *complessità del ragionamento* che il sistema deve eseguire per rispondere a queste domande.

Gli elementi che contribuiscono a definire la complessità del simulatore sono l'*obiettivo* del dialogo e l'*iniziativa* (del sistema o dell'utente).

Facciamo qualche esempio.

## Un dialogo molto semplice:

Information-giving, system initiative, formal attitude

S: Wellcome to Irene's travel agency. I'm here to help you to buy your travel tickets. Which place do you want to go?

U: London

S: Which place do you start from?

U: Bari

S: Do you prefer travelling with a regular or a low-cost company?

U: Regular

S: Which tarif do you prefer? Economic or business?

.....

**Qui, S guida il dialogo, e le risposte di U appartengono a categorie predefinite.**

**Non è necessario un parser complesso né una interpretazione della mossa di U.**

**Non sono necessarie revisioni di goal e di piani, ma al massimo piccoli aggiustamenti.**

## Un dialogo più complesso:

Information-giving, mixed initiative, friendly attitude

S: Wellcome to Irene's travel agency. I'm here to help you to organize your travels.

U: I would like to go to London starting from Bari in the month of may, and have not much money to spend.

S: Oh, London! I would love going there! There are several low-cost companies going there. Let's see what we can do. Do you mind to transfer from Fiumicino to Ciampino, when you are in Rome?

U: No problem: I'm young and used to travel.

S: Good, no problem then. Just tell me when exactly you would like to leave.

.....

**Qui, S ha un doppio goal: dare informazioni e 'stabilire un clima di cooperazione' con U.**

**Il linguaggio d'interazione di U è abbastanza rigido, e le mosse che può fare sono predefinite.**

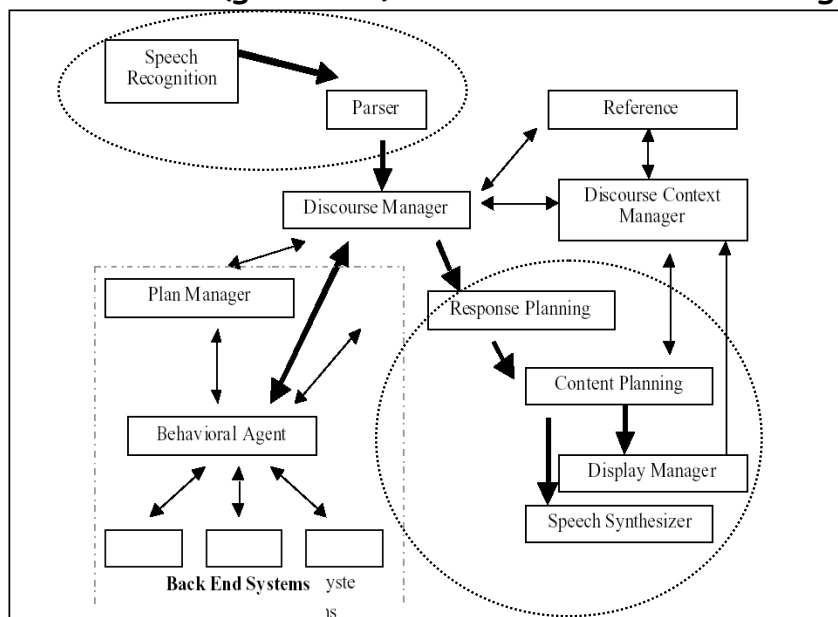
## Un dialogo ancora più complesso: information-giving, mixed initiative

S: Dovresti andare a correre, Giuseppe! *Prima mossa persuasiva*  
 U: Perché?  
 S: Perché sei giovane, ci tieni alla tua salute, e correre fa bene alla salute. *Linea di ragionamento seguita*  
 U: Ma ti pare che, a vent'anni, già devo pensare alla salute?  
 S: Secondo me sì. Ma comunque, correre aiuta anche a tenersi in forma. *Correzione della strategia persuasiva*  
 U: E chi te l'ha detto?  
 S: Lo dicono studi epidemiologici svolti in diversi paesi, da istituti di ricerca qualificati. *Giustificazione di un passo di ragionamento*  
 U: Ma io detesto correre. *ragionamento*  
 ...

**Qui, ad ogni passo U può fare mosse di tipo diverso:  
chiedere più informazioni,  
verificare come S sta ragionando,  
obiettare, ...**

9

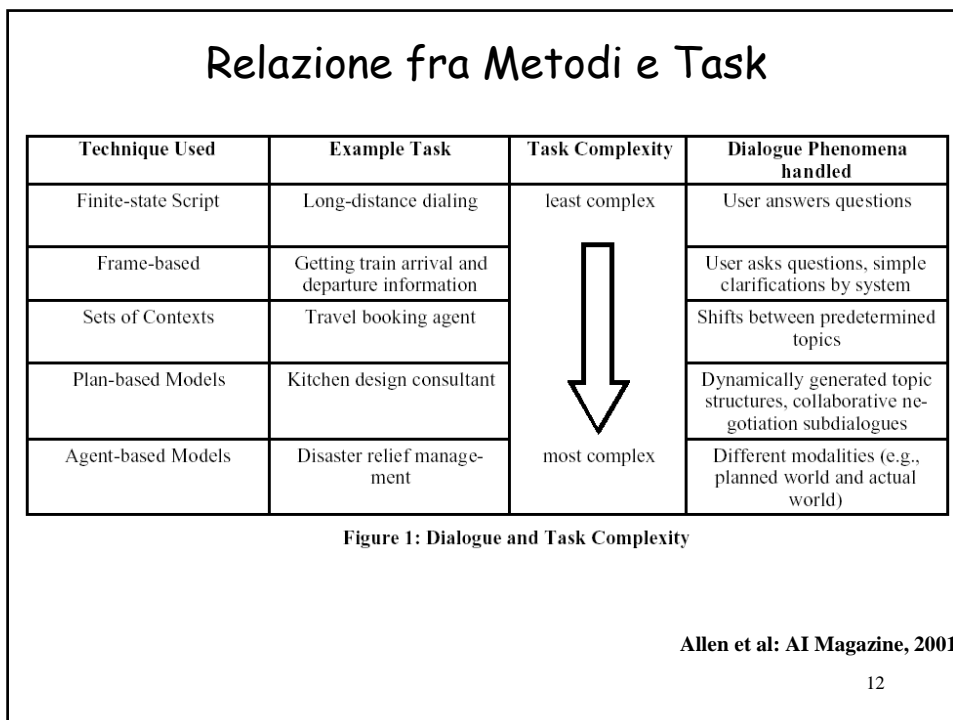
## Architettura (generale) di un simulatore di dialoghi



*Figure 1: The Abstract Architecture of the Dialogue Shell*

Module	Function	<i>Ruolo delle diverse componenti</i>
Speech Recognition (SR)	Transforming speech input into a word stream or word lattice	
Parser	Transforming the SR output into interpretations, each a set of conventional speech acts, using full and robust parsing techniques	
Reference Manager (REF)	Identifying the most salient referents for referring expressions such as noun phrases	
Discourse Context Manager	Maintaining the global (topic flow) and local (salience with a topic) discourse context	
Discourse Manager (DM)	Identifying the intended speech act, current task, current step in the current task, and system obligations arising from the dialogue	
Behavioral Agent (BA)	Determines system actions (e.g., answer a question, notify of a problem, request clarification); Manages the interface to the back-end systems.	
Plan Manager	Constructing, modifying, evaluating, and executing plans (whether they are the subject of the conversation or the task being executed)	
World KB	Maintains a description of the current state of the world under differing assumptions (e.g., based on different plans or hypotheses)	
Response Planner	Determining the best communicative act(s) (and their content) to accomplish the system's current goals and discourse obligations	
Content Planner	Determining how to realize the planned speech acts	
Display Manager	Managing the visual presentations given the available hardware.	

*Table 2: The Key Modules in the Dialogue Shell in the Abstract Architecture*



Vedremo ora due dei metodi principali per la simulazione di dialoghi:

- a. Simulatori a stati finiti e
- b. Simulatori basati sull'Information state, confrontandone vantaggi e svantaggi.

13

### Simulatori a Stati Finiti, eventualmente 'aumentati'

(ATN = Augmented Transition Networks)

Sono utilizzabili quando l'iniziativa è del sistema, che segue un *piano predefinito* per raccogliere informazioni dall'utente.

Gli stati rappresentano il risultato dell'esecuzione di una mossa di dialogo nello stato precedente.

Per ogni stato, viene definita una lista di possibili mosse successive.

Ad ogni stato sono associate delle 'azioni' che il sistema esegue.

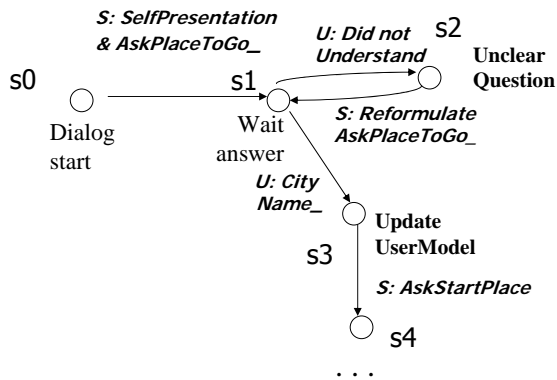
Il passaggio da uno stato all'altro è reso possibile dal verificarsi di una serie di condizioni, legate alla mossa eseguita dall'utente e al valore di alcuni registri.

... vediamo qualche esempio ...

14

# Una ATN molto semplice

per il dialogo della Travel Agency di Irene



S: Wellcome to Irene's travel agency. I'm here to help you to organize your travels. Which place do you want to go?

U: London

S: Which place do you start from?

U: Bari

S: Do you prefer travelling with a regular or a low-cost company?

U: Regular

S: Which tarif do you prefer? Economic or business?

Vediamo ora un esempio di sistema basato su ATN.  
**Obiettivo: dare suggerimenti sullo sport utilizzando un Agente Animato Conversazionale**

# Il Sistema FitTrack dell' MIT

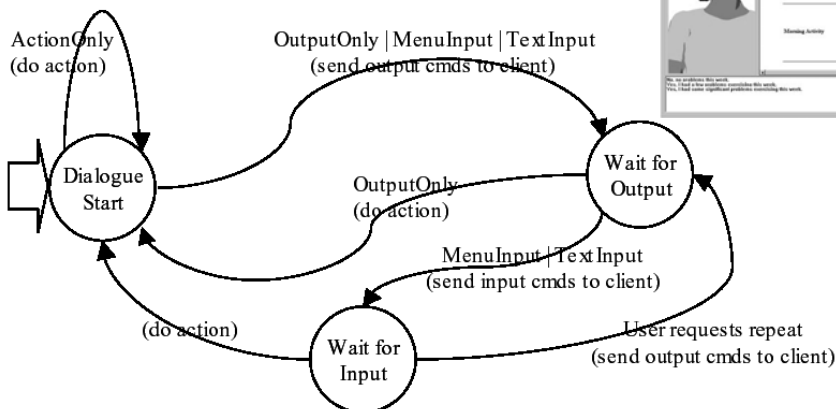


Figure 8-7. Fragment of ATN Executive State Machine

Una macchina a stati finiti + uno *stack*.

L'esecuzione può passare da un nodo all'altro, oppure al nodo iniziale di un'altra ATN, inserendo il nome del nodo attuale nello *stack* (*innestamento*).



## Lo Scripting Language di FitTrack

Nella forma più semplice, ogni arco dell'ATN viene rappresentato mediante una regola del tipo:

STATE: Si

Tag per l'animazione del character

AGENT: \$ <BEAT>Are you going to work out tomorrow?</BEAT>\$

USERMENU: I can't. => \$ Go("Sj"); \$

Yep. => \$ Go("Sk"); \$

Oltre ad azioni di Go, è possibile inserire altre funzioni, di azione sullo stack (PUSH e POP), per realizzare l'innestamento fra ATN.

*... un compilatore produce l'eseguibile...*

Questo rende molto flessibile  
la rappresentazione della dinamica del dialogo

17

## FitTrack: Esempi per diverse 'personalità' dell'agente

### "Int1" script

```
STATE: Int1
AGENT_REL: $ <BEAT> <HAPPY/> Hi there. To talk to me, <TEXTBOX> just click on one
of the options in the menu below </TEXTBOX>. <NEUTRAL/> </BEAT> $
AGENT_NREL: $ <BEAT> <NEUTRAL/> Hi there. To talk to me, <TEXTBOX> just click on
one of the options in the menu below </TEXTBOX>. </BEAT> $
REPEAT: $ <BEAT>Just click the mouse below where it says OK.</BEAT> $
USERMENU: OK, I understand that this is where I click. => $ GO("INT1_1"); $

STATE: INT1_1
ACTION: $ if(ISREL()) GO("INT1_3"); else GO("INT1_2"); $

STATE: INT1_2
AGENT: $ <BEAT> Good. I'm going to be your exercise advisor. </BEAT> $
ACTION: $ GO("INT1_11"); $

STATE: INT1_3
AGENT: $ <BEAT> Good. My name is Laura. <YOU> You are =| GET("NAME") |= , is that right?
</YOU> </BEAT> $
USERMENU: Yep, that's right. => $ GO("INT1_5A"); $
No, actually that's not right. => $ GO("INT1_4"); $

STATE: INT1_4
AGENT: $ <BEAT> Oh, I'm sorry. What is your name? </BEAT> $
USERTEXT: My name is: => $ String entry=GETTEXT();
ALERT("WRONG NAME: not "+GET("NAME")+ " corrected to "+entry);
entry=FILTERNAME(entry);
if(entry.equals("NOTCHECKED")) entry="";
SET("NAME",entry);
GO("INT1_5"); $
```

<b>Script Primitives</b>		<b>Operazioni su Registri</b>
<b>Type</b>	<b>Function</b>	<b>Purpose</b>
<b>Property Access Functions</b>	GET(<property>)	Looks up single-valued property value in database, returned as a String.
	GETINT(<property>)	Returns value as an integer.
	UNDEFINED(<value>)	Returns 'true' if the property value is undefined.
	SET(<property>,<value>)	Sets a single-valued property in the database.
	GET_SESSION(<property>,<index>)	Looks up historical property value. If (integer) index is positive, this returns the value for the specified session. If index is negative, this uses relative indexing (e.g., "-1" returns the value from the previous session). If zero, this returns the value from the current session.
	GET_SESSION(<property>)	Looks up historical property value for the current session.
	SET_SESSION(<property>,<value>)	Sets a historical property value, indexed to the current session.
	EXISTS_SESSION(<property>,<value>)	Returns most recent session index in which <property> had <value>.
	COUNT_SESSION(<property>,<value>)	Returns the number of sessions in which <property> had value <value>.
	GET_EQ(<property>,<value>)	Returns 'true' if single-valued <property> had value <value>.
	GET_SESSION_EQ(<property>,<value>)	Returns 'true' if historical <property> has value <value> in current session.
	GET_SESSION_EQ(<property>,<index>,<value>)	Returns 'true' if historical <property> has value <value> in the specified session.

19

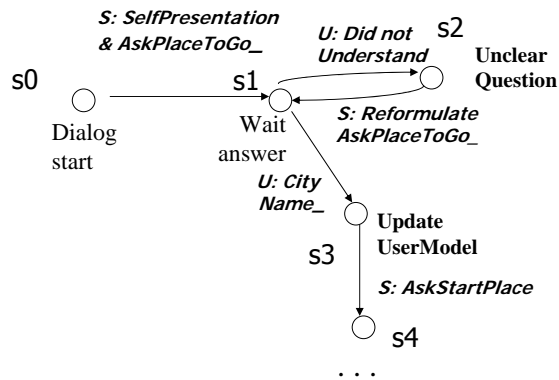
**(dalla Tesi di PhD di T. Bickmore)**

<b>... Continuano le funzioni di FitTrack</b>		
<b>ATN State Change Functions</b>	GO(<state>)	Branches to the named state.
	PUSH(<nextState>,<returnState>)	Pushes <returnState> on the runtime stack and branches to <nextState>.
	POP()	Pops the runtime stack and branches to the returned state.
<b>User Interface Functions</b>	GETTEXT()	Returns the text entered by the user.
	CONTENT(<n>)	Displays educational content page for session <n> in the main browser window.
<b>Database lookup</b>	ISREL()	Returns 'true' if user is in RELATIONAL condition of the study, else 'false' (if in NON-RELATIONAL).
	NAME()	Returns subject's given name.
<b>Script Primitives, continued</b>		
<b>Type</b>	<b>Function</b>	<b>Purpose</b>
<b>Miscellaneous</b>	PICK(<n>)	Returns a random number between 1 and <n>.
	REFER_SESSION(<n>)	Returns an English expression referring to the time of the specified session (e.g., "YESTERDAY", "LAST TUESDAY", etc.)
	GOALTIME()	Returns the number of minutes the user should be exercising today.
	GOALSTEPS()	Returns the number of steps the user should be walking today.
	ALERT(<string>)	Enters message into the Alert table in the database (reviewed daily by operator for exceptions).

20

## Riprendiamo la nostra ATN

per il dialogo della Travel Agency di Irene



**S:** Wellcome to Irene's travel agency. I'm here to help you to organize your travels. Which place do you want to go?

**U:** London

**S:** Which place do you start from?

**U:** Bari

**S:** Do you prefer travelling with a regular or a low-cost company?

**U:** Regular

**S:** Which tarif do you prefer? Economic or business?

21

## Esercizio 1

Prova a simulare come varia, ad ogni passo, il modello dell'utente costruito da S, nel dialogo precedente.

22

## Traduzione dell'ATN in uno scripting language

STATE: S0

AGENT: \$<BEAT><SMILE/> **Wellcome to Irene's travel agency. I'm here to help you to organize your travels. Which place do you want to go?**</ </BEAT>\$  
\$ Go("S1"); \$

STATE: S1

USERMENU: if "Can you please repeat?" => \$ Go("S2") \$;  
else \$ string entry = GETTEXT()  
SET ("DEPLACE(U)" = entry) \$ => \$ Go("S3");

STATE: S3

AGENT: \$<BEAT>**Which place do you want to start from?**  
</BEAT>\$ => \$ Go("S4") \$;

23

## Introduciamo un adattamento del dialogo alla lingua

### **1° possibilità:**

introdurre nello script tante varianti della 'agent move' quante sono le lingue in cui il sistema deve essere in grado di parlare.

STATE: S0

AGENT\_ENGL: \$<BEAT><SMILE/>**Benvenuta nell'Agenzia di viaggio di Irene!. Sono qui per aiutarti ad organizzare il tuo viaggio. Dove vuoi andare?**</BEAT>\$

AGENT\_ENGL: \$<BEAT><SMILE/>

**Wellcome to Irene's travel agency. I'm here to help you to organize your travels. Which place do you want to go?**</BEAT>\$  
\$ Go("S1"); \$

24

### 2° possibilità:

introdurre nello script una funzione di lettura della frase che l'agente deve pronunciare (in forma simbolica), da un DB associato.

```
STATE: S0
GET ("LANG", I)
AGENT: $ <BEAT> <SMILE/> GETTEXT(SelfPresentation(A, I));
      GETTEXT(AskPlaceToGo(A,I)) </BEAT> $
      => $ Go("S1"); $

STATE: S1
USERMENU: GETTEXT(AskToRepeat(U,I)) => $ Go("S2") $;
          $ string entry = GETTEXT();
          SET ("GOTOPLACE(U)" = entry) & =>
          $ Go("S3") $;

STATE: S3
AGENT: $ <BEAT> GETTEXT(AskPlaceToStart(A,I)) </BEAT> $
      => $ Go("S4") $;
```

25

### Ho utilizzato due registri e un DB

LANG            nome della lingua  
GOTOPLACE    nome della città in cui andare

↑  
*Questo è parte dello User Model*

Per denotare le frasi che l'Agente può pronunciare, utilizzerò una qualche classificazione di atti comunicativi (Searle o altri) e leggerò i loro valori, nella lingua voluta, da un DB:

SelfPresentation (A, ENGL)	Wellcome to Irene's travel agency! I'm here to help you to organize your travels.
SelfPresentation (A, ITAL)	Benvenuta nell'Agenzia di viaggio di Irene! Sono qui per aiutarti ad organizzare il tuo viaggio.
... ecc	

**Qual è la differenza fra registri e DB???????**

26

### **3° possibilità:**

introdurre nello script una chiamata ad una funzione di generazione di frasi in linguaggio naturale, a partire da una descrizione in forma simbolica della frase che l'agente deve pronunciare, oppure da un goal comunicativo.

```
STATE: S0
GET ("LANG", I)
AGENT: $<BEAT><SMILE/>GENERATE(SelfPresentation(A, I));
      GENERATE(AskPlaceToGo(A,I))</BEAT>$
      => $ Go("S1"); $

STATE: S1
USERMENU: GENERATE(AskToRepeat(U,I)) => $ Go("S2") $;
          $ string entry = GETTEXT();
          SET ("GOTOPLACE(U)" = entry) & =>
          $ Go("S3") $;

STATE: S3
AGENT: $<BEAT> GENERATE(AskPlaceToStart(A,I))</BEAT>$
      => $ Go("S4") $;
```

27

## **Che succede se voglio simulare l'esempio un po' più complesso?**

S: Wellcome to Irene's travel agency. I'm here to help you to organize your travels.

**U: I would like to go to London starting from Bari in the month of may, and have not much money to spend.**

S: Oh, London! I would love going there! There are several low-cost companies going there. Let's see what we can do. Do you mind to transfer from Fiumicino to Ciampino, when you are in Rome?

U: No problem: I'm young and used to travel.

S: Good, no problem then. Just tell me when exactly you would like to leave.

.....

28

## Traduzione dell'ATN in uno scripting language

STATE: S0

AGENT: \$<BEAT><SMILE/> Wellcome to Irene's travel agency. I'm here to help you to organize your travels. Which place do you want to go?</ </BEAT>\$  
\$ Go("S1"); \$

STATE: S1

USERMENU: "Can you please repeat?" \$ Go("S2"); \$  
\$ string entry = GETTEXT()  
????\$ Go("???"); \$

**DEVO INTRODURRE QUI UNA FUNZIONE DI PARSING DELLA STRINGA DI INPUT, CHE INTERPRETI COSA L'UTENTE HA DETTO E AGGIORNI I REGISTRI CORRISPONDENTI.**

Ma devo anche saltare a stati diversi, a seconda delle informazioni che U ha fornito (S3 se ha indicato solo la città di arrivo, S4 se ha indicato anche quella di partenza)

29

## ... ma non basta!...

S: Wellcome to Irene's travel agency. I'm here to help you to organize your travels.

U: I would like to go to London starting from Bari in the month of may, and have not much money to spend.

S: Oh, London! I would love going there! There are several low-cost companies going there. Let's see what we can do. Do you mind to transfer from Fiumicino to Ciampino, when you are in Rome?

**U: No problem: I'm young and used to travel.**

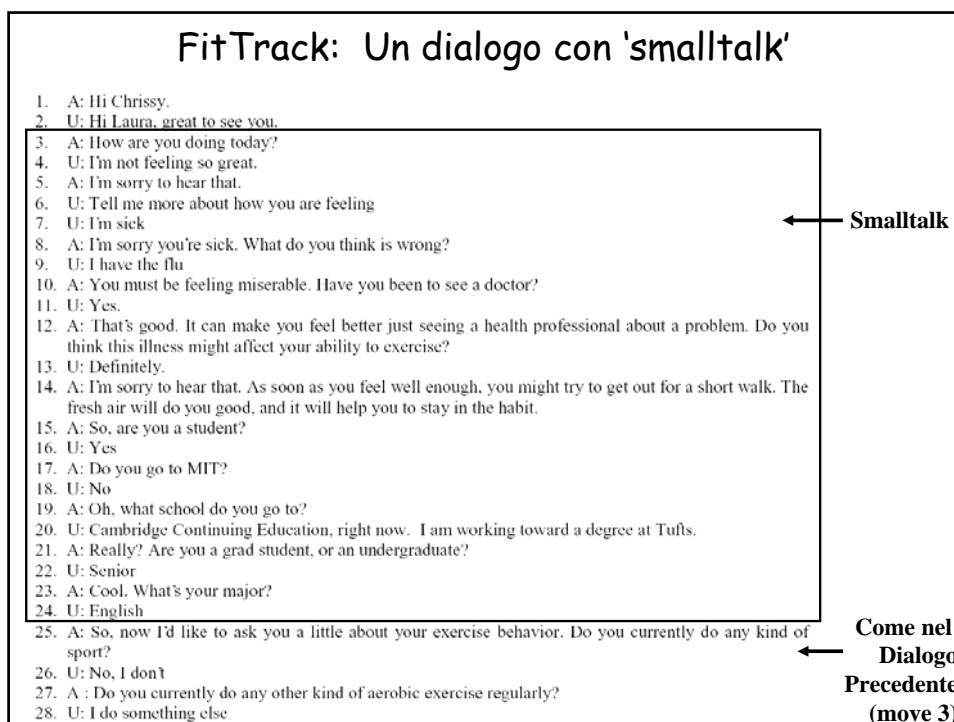
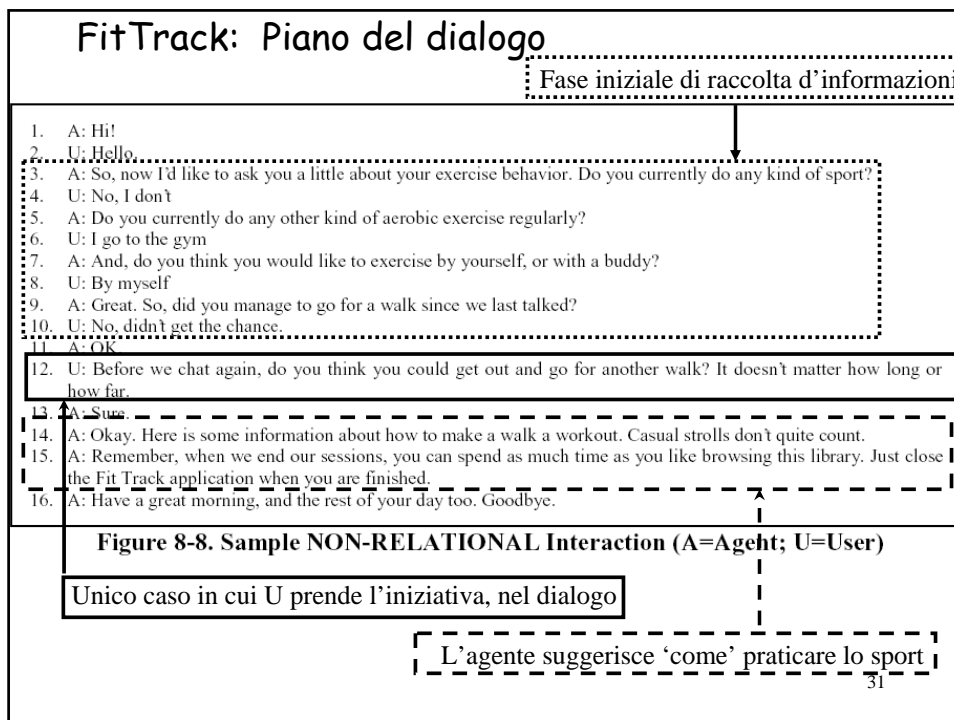
S: Good, no problem then. Just tell me when exactly you would like to leave.

.....

**IN QUESTO CASO, LA FUNZIONE DI PARSING DELLA STRINGA DI INPUT NON BASTA: OCCORRE CONCATENARE QUESTA UNA FUNZIONE DI *INFERENZA* (eventualmente in condizioni di *incertezza*) DI ALTRE INFORMAZIONI**

Young(U)  $\wedge$  UsedToTravel(U)  $\rightarrow$  PrefersLowCost(U)

30





## FitTrack: Piano del dialogo

Goal: to suggest an appropriate sport

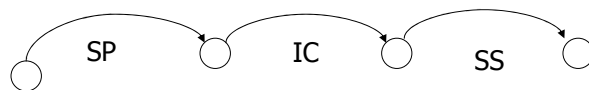
Subgoals:

self\_presentation (SP)

information collection (IC)

sport\_suggestion (SS)

*Sviluppo di ogni sottopiano mediante innestamento di ATN*



Quando e come inserire smalltalk?

- in una fase prestabilita dal sistema, oppure
- durante l'esecuzione di uno dei sottopiani, se l'utente mostra un atteggiamento 'amichevole', dimostrandosi disponibile alla 'chiacchiera'. In questo caso, il sistema dovrà memorizzare in un registro la 'storia del dialogo', e cioè l'elenco dei diversi stati attraverso i quali è passato. 33

## Esercizio 2

Prova a simulare con un simulatore a stati finiti una parte del dialogo di travel agency più complesso, introducendo una funzione di innestamento e la possibilità di smalltalk su iniziativa dell'utente.

## Vantaggi e limiti dei metodi script-based

Il numero di stati attraverso i quali il sistema può passare è finito e predeterminato.

La logica di sviluppo del dialogo è espressa in modo molto chiaro

L'iniziativa è fondamentalemente del sistema

Gli utenti devono rispondere con il livello di dettaglio e l'ordine di presentazione delle informazioni previsti dal sistema

E' difficile realizzare un sistema che adatta il suo comportamento alle caratteristiche dell'utente (utilizzando un modello del suo stato mentale)

35

## Dialoghi nei Call Center

### **Idea di fondo:**

Realizzare il dialogo in modo automatico 'finché tutto va bene', dotando il sistema della capacità di capire il più possibile quello che l'utente desidera, *interpretando le sue richieste in modo 'intelligente'*.

Quando il sistema si rende conto che 'qualcosa non va', passare ad un operatore umano.

**(uno dei settori di applicazione dei dialoghi speech-based di maggiore interesse, sia a livello di ricerca che a livello industriale)**

36

## Un Esempio: ARTIMIS (France Telecom)

S: Welcome to the ARTIMIS server. How can I help you?

U: Hello, I would like to know the sea weather forecast for the Finistère area.

S: There is no sea weather forecast server for the Finistère area.

But there are some for the Channel and for the Atlantic.  
Are you interested in one of these?

U: Yes, mmm the Channel

S: There are two servers for coastal weather forecast for the Channel.  
Their phone numbers are.... Would you like more information?

U: Does it have specific weather forecast for sailing?

S: No it doesn't.

There is no weather forecast specialized for sailing for the Channel.  
But there are some for France. Is that OK?

U: Yes.

..... **S capisce qual è l'obiettivo di U e non si limita a rispondere alla sua richiesta, ma cerca di dare una risposta adeguata a questo obiettivo.**

37

**Per formalizzare l'esempio precedente,  
ricordiamo la Classificazione di Searle**  
degli atti comunicativi, in funzione degli 'obiettivi comunicativi'

*Constativi:*

- informare, avvertire  
*esprimono il desiderio che H creda in un determinato fatto*
- affermare, annunciare, rispondere, sostenere, confermare, negare,  
...  
*esprimono il desiderio che H creda che S crede in un determinato fatto*
- chiedere, domandare:  
*esprimono il desiderio di conoscere il pensiero di H su un determinato fatto*

*Direttivi:* suggerire, ordinare, istruire, permettere,...

*esprimono il desiderio che H compia una determinata azione*

*Commissivi:* offrire, promettere, garantire, ...

*esprimono l'intenzione, da parte di S, di compiere una determinata azione*

*Riconoscitivi:* scusarsi, congratularsi, ringraziare, accettare,...

*coinvolgono il passaggio d'informazioni su 'atteggiamenti di gentilezza' di S*

38

## Elaborazione degli atti comunicativi di U

Analizziamo le 'mosse' di U nel dialogo precedente, formalizzandole in termini dei seguenti atti comunicativi:

x variabile che denota un information item; a variabile che denota un'azione;  $\phi$  formula; ag variabile che denota S/U

### *Constativi*

Inform / AskInfo (ag x)

Answer / Announce / Confirm / AskIf (ag  $\phi$ )

### *Direttivi*

Request / Order (ag a)

### *Commissivi*

Offer / Promise (ag a)

### *Riconoscitivi*

Apologize / Congratulate / Thank / Acknowledge (ag  $\phi$ )

39

## Elaborazione degli atti comunicativi di U

U: Hello, I would like to know the sea weather forecast for the Finistère area.

a) *Riconoscimento*: traduzione della frase in LN in forma simbolica:

*U ha chiesto informazioni sulle previsioni del tempo nel Finistère*

AskInfo(U, WeFo<sup>o</sup>(Finistere))

b) *Interpretazione*: ragionamento sullo stato mentale di U che ha prodotto quell'atto comunicativo:

*U vuole conoscere le previsioni del tempo nell'area*

(Bel S WantToKnow(U, WeFo<sup>o</sup>(Finistere))) \*

c) *Ragionamento su di sé* (come rispondo alla sua mossa?)

\* ... ricordiamo le ipotesi sull'atto comunicativo di AskInfo...

40

## Ragionamento sull'atto comunicativo di U

- c) Ragionamento su di sé (come rispondo alla sua mossa?)  
 (Bel S WantToKnow(U, WeFo°(Finistere))):  
**KnowAbout(S, WeFo°(Finistere)) ? No!**  
*S non le conosce; ma sa che:*  
**Is-in(Finistere, NorthernFrance)**  
**KnowAbout(S, WeFo°(NorthernFrance))**  
*Il Finistere è nel nord della Francia*  
*e S conosce le previsioni in questa zona; quindi conclude che (forse):*  
**(Bel S WantToKnow(U, WeFo°(NorthernFrance)))**  
*S crede che U voglia conoscere le previsioni nel nord della Francia*  
**IsIn(Channel, NorthernFrance)**  
**(Bel S WantToKnow(U, WeFo°(Channel)))**  
*In particolare, la Manica è nel nord della Francia*  
*S crede che U voglia conoscere le previsioni nella Manica*

**Regola applicata da S (ragionamento su U, su S e sul dominio)**  
 Bel S (WantToKnow (U,x)  $\wedge$   $\neg$ KnowAbout(S,x)  $\wedge$  (Similar(x,y)  $\vee$  IsIn(x,y))  $\wedge$  KnowAbout(S,y)  $\rightarrow$  WantToKnow(U,y))

## Traduzione del ragionamento nella mossa di dialogo successiva

Piano che combina:

Answer (S  $\neg$ KnowAbout(S, WeFo°(Finistere)))

e

Answer(S KnowAbout(S WeFo°(Channel)))

... con una relazione retorica di 'Contrast'...

e

OfferInfo (S WeFo°(Channel))

Con una 'Joint'

Per giustificare perché non risponde direttamente alla domanda di U



Per verificare se la sua ipotesi è corretta



S: There is no sea wheather forecast server for the Finistere area. But there are some for the Channel and for the Atlantic.

Are you interested in one of these?

(traduzione in LN della Answer e della Offer)

## Cosa occorre per gestire questo tipo di dialogo?

### **Fonti d'informazione:**

Un *modello dell'utente*, in cui memorizzare le informazioni acquisite (o inferite) durante il dialogo.

Una base di *conoscenza sul dominio*.

Una *agenda* delle cose da fare.

Una *'storia' del dialogo*, e in particolare dell'ultima 'mossa' (chi ha parlato e cosa ha detto, quale piano ha applicato il sistema, ecc).

### **Capacità di ragionamento:**

**Interpretare la mossa dell'utente.**

**Aggiornare il suo modello garantendone la consistenza.**

**Scegliere il piano successivo ed eseguirlo.**

**Tradurre il piano in linguaggio naturale o in forma multimediale.**

43

## Una soluzione possibile per simulare questo tipo i dialoghi:

una 'Lavagna'

E' necessario uno spazio di memoria (una 'blackboard')  
in cui memorizzare i risultati del ragionamento,  
con una differenziazione fra fatti acquisiti ('shared') e fatti inferiti  
(*'private'*):

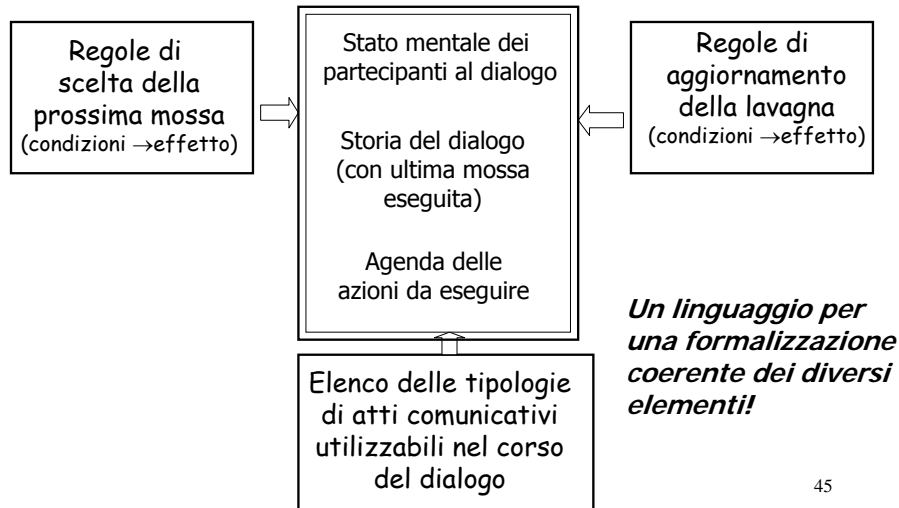
Su U: WantToKnow(U, WeFo°(Finistere))	shared
WantToKnow(U, WeFo°(NorthernFrance))	private
WantToKnow(U, WeFo°(Channel))	private

Nella stessa blackboard si può registrare la 'storia del dialogo':

AskInfo(U, WeFo°(Finistere))  
Answer (S, ¬ KnowAbout(S, WeFo°(Finistere)))  
Answer (S KnowAbout(S, WeFo°(Channel)))  
OfferInfo (S, WeFo°(Channel))

44

## Una Lavagna 'attiva' (Information State)



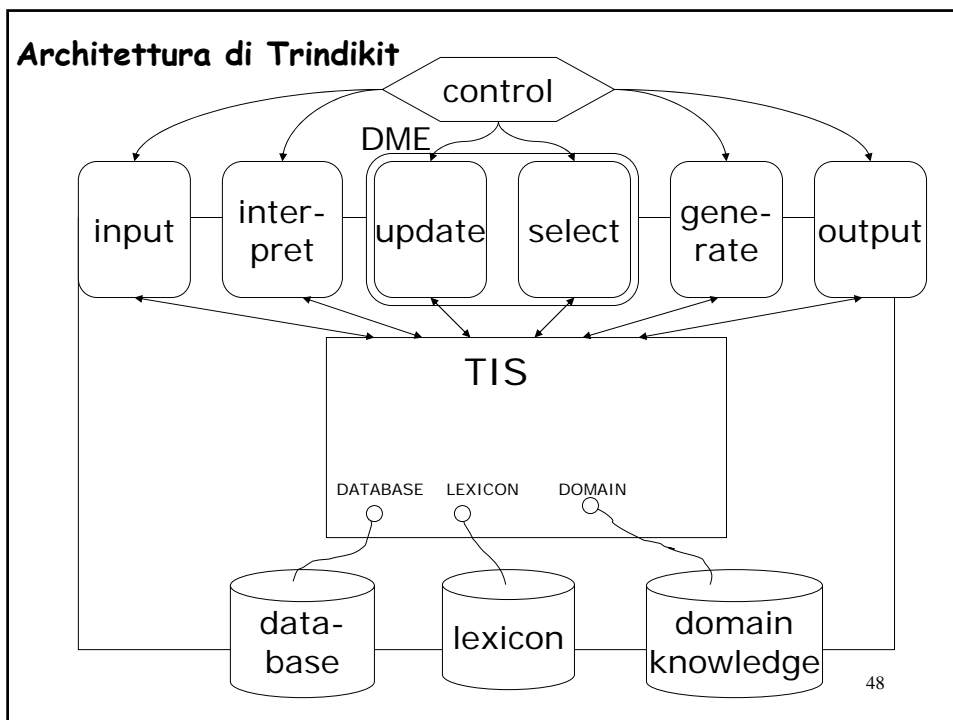
## Approccio al Dialog Management basato sull' Information State

- *Information State (IS)*
  - an abstract data structure (record, DRS, set, stack etc.)
  - accessed by dialogue system modules using conditions and operations
- *Dialogue Moves*
  - utterance function (ask, answer, request etc.)
- *Update rules*
  - Modify IS based on on observed moves
  - Select moves to be performed
- IS Approach implemented in TrindiKit

# Trindikit

Un simulatore di dialoghi in Prolog basato sul metodo dell'information state, sviluppato nell'ambito di un Progetto Europeo.

47





## La struttura dell'IS

PRIVATE	:	BEL	: Set(Prop)
		AGENDA	: Stack(Action)
SHARED	:	BEL	: Set(Prop)
		QUD	: Stack(Question)
		LM	: Move

BEL: conoscenza acquisita durante il dialogo

AGENDA : stack delle azioni da eseguire

QUD : questions under discussion

LM: latest dialogue move performed

PRIVATE : conoscenza inferita dal sistema

SHARED : conoscenza esplicitamente dichiarata durante il dialogo

L'IS rappresenta quindi, in particolare, lo User Model, la storia del dialogo e un'agenda delle cose da fare, mentre la conoscenza sul dominio è rappresentata in una struttura separata.

49

## Update rule

Permettono di aggiornare l'IS dopo una 'mossa' di S o di U.

*Modello di Utente:*

- dopo un atto comunicativo CA di U: aggiornamento dell'UM con gli scopi di CA(U); *(perché U ha detto CA(U)?)*
- dopo un atto comunicativo di S: aggiornamento dell'UM con gli effetti di CA(S); *(quali effetti presumibilmente ha prodotto CA(S)?)*

**Esempio:**

(AskInfo U,x) → WantToKnow(U,x)

(Inform U, φ) → Bel U φ

*Ma l'aggiornamento può riguardare anche le implicazioni di quanto U o S hanno detto.*

*Ad esempio, in un lucido precedente avevamo:*

(WantToKnow (U,x) ∧ (Similar(x,y) ∨ IsIn(x,y))) → WantToKnow(U,y)

50

## Select rule

Permettono ad S di selezionare la prossima 'mossa' da compiere (singolo atto comunicativo o piano), sulla base del contenuto dell'IS (UM, stato del dialogo, obiettivi che S intende raggiungere).

Esempio: **Answer directly**

$(\text{WantToKnow}(U,x) \wedge \text{KnowAbout}(S,x)) \rightarrow \text{InformAbout}(S,x)$

Esempio: **Answer indirectly**

$(\text{WantToKnow}(U,x) \wedge \neg \text{KnowAbout}(S,x) \wedge (\text{Similar}(x,y) \vee \text{IsIn}(x,y))) \wedge \text{KnowAbout}(S,y) \rightarrow \text{InformAbout}(S,y)$

Esempio: **OpenSmallTalk \***

$(\text{LM}(U, \text{Greet}) \wedge (\text{Bel } S \neg \text{Socialized}(U)) \wedge (\text{Goal } S \text{ Socialized}(U))) \rightarrow \text{NextPlan}(S, \text{SmallTalk})$

\* Apre un dialogo di smalltalk all'inizio dell'interazione, come nell'esempio di FitTrack

51

## Struttura di controllo

```
control_algorithm( [ ! reset,
    * [ * [ ( ! select ,
        ( ( or( or( or( fstRec(private^agenda, ifem( _,_ ) ), fstRec(
private^agenda, if( _,_ ) ) ), fstRec( private^agenda, ifthen( _,_ ) ) ), fstRec(
private^agenda, end ) ) , ! [ ! adequate, ! update, ! goal_check, ! print_state ] );
        ! [ ! generate,
            ! output,
            ! update,
                ! goal_check,
            ! print_state ] ) ) ,
        ( or( emptyRec( private^agenda ) ,
            fstRec( private^agenda, raise( _^(task=_ ) ) ) ) ) ,

        program_state:val(run),
        ! input,
            ! interpret,
            ! update,
        ! print_state
    ] ,
    program_state:val(run)
] ).
```

52

## Struttura di controllo

```
control_algorithm
While system_goals = empty:
  Select next move from agenda
  if nil, then
    select next goal; read nextplan
    add (nextplan, agenda)
  Generate(car(nextplan), system_move)
  Output system_move
  Read user_input
  Interpret (user_input, user_move)
  Update IS: add(user_move, dialog_history, )
  Update IS: integrate(user_move, user_model)
  Update IS: revise(system_goals)
end
```

53

## Generatore

Traduce le singole mosse che il sistema ha stabilito di compiere  
(rappresentate in forma simbolica)  
in frasi in linguaggio naturale  
oppure frasi annotate in un linguaggio di markup,  
applicando uno dei metodi  
che abbiamo esaminato, in breve, nelle unità precedenti.

54

## Vediamo un esempio di dialogo con un Agente animato simile a FitTrack

S1: Dovresti andare a correre, Giuseppe!

U1: Perché?

S2: Perché ci tieni alla tua salute e correre fa bene alla salute.

U2: *Ma tu ci vai, a correre?*

S3: *Io sono un agente artificiale, non posso correre!*

U3: *Oh peccato, mi sarebbe piaciuto fare una corsetta con te!*

S4: *Grazie, sei gentile. Chissà, in un mondo virtuale! Per ora, puoi andare a correre con un'amica, se vuoi restare in buona salute...*

U4: Ma ti pare che a ventanni devo già pensare alla salute?

S5: Ma correre tiene anche in forma!

55

## Dialogue games

L'esempio precedente fa riferimento alla categoria dei *dialogue games*, in cui (in una certa fase del dialogo) *i due interlocutori hanno goal diversi*:

S vuole persuadere U a fare sport:

U vuole scherzare con S.

S potrebbe rifiutare il goal di U. Ma invece adotta un atteggiamento 'aperto'. Lo accetta temporaneamente (in S3 e S4), per poi riprendere il suo (nella stessa S4).

Vediamo la dinamica del contenuto dell'IS in questo dialogo.

Ne descriveremo una possibile simulazione introducendo nell'IS una agenda e senza formalizzare né la rappresentazione della conoscenza né il suo trattamento.

56

Piano iniziale di S: *Persuadere U motivando*:

```
<DPL><plan name="RatPers">
  <goal name="Persuade" term="S-Do(Mary Run)" RR="Motivation"> ← Subgoal1
  < communicative_act name = "Suggest" term=" S-Do(Giuseppe Run)" type="N">
  </communicative_act>
    <goal name="DescribeWhy" term="S-Do(Giuseppe Run)" RR="Joint"> Subgoal2
    <communicative_act name="Remind" term="V-Goal(Giuseppe
    GoodHealth)">
    </communicative_act>
    <communicative_act name="Inform" term="Implies(Run
    GoodHealth)">
    </communicative_act>
  </goal>
</goal>
</plan></DPL>
```

S mette il piano in una 'agenda' nell'IS, gestita come stack:

```
(communicative_act name =Suggest, term=S-Do(Giuseppe Run))
((communicative_act name=Remind, term=V-Goal(Giuseppe GoodHealth));
 (communicative_act name=Inform, term=Implies(Run GoodHealth)))
```

Inizia ad eseguire il suo piano, considerando il primo elemento dello stack:

S1: Dovresti andare a correre, Giuseppe!

Cede il controllo al generatore superficiale, che inserisce i tag APLM nella 'mossa di dialogo' e invia l'input al player del character animato.

Passa il controllo al modulo di *input interpretation* e osserva la reazione dell'utente.<sup>57</sup>

U1: *Perché?*

Input Interpretation interpreta la User move come richiesta di giustificazione del suggerimento.

Questo attiva il Subgoal2 di S, che determina l'esecuzione delle altre due componenti dell'agenda:

```
((communicative_act name=Remind, term=V-Goal(Giuseppe GoodHealth));
 (communicative_act name=Inform, term=Implies(Run GoodHealth)))
```

S2: *Perché ci tieni alla tua salute e correre fa bene alla salute.*

L'agenda di S, ora, è vuota. U risponde:

U2: *Ma tu ci vai a correre?*

Input Interpretation interpreta la User move come AskWhether; interroga la sua base di conoscenza sull'Agente; scopre di non poter correre perché è un agente artificiale.

Applicando una select-rule, pone la sua risposta in cima all'agenda:

```
((communicative_act name=Answer, term=notCan(S,run));
 (communicative_act name=Inform, term=ArtificialAgent(S)))
```

S3: *Io non posso correre, sono un agente artificiale!*

Ora U risponde:

U3: *Oh peccato, mi sarebbe piaciuto fare una corsetta con te!*

Input Interpretation interpreta la User move come 'SocializationAttempt'.

58

Applicando una select-rule, decide di rispondere ricambiando la gentilezza e riprendendo il suo suggerimento iniziale, adattato al contesto del dialogo.

Pone in cima alla sua agenda una risposta:

**(communicative\_act name=Acknowledge, term=Kind(U));**

**(communicative\_act name=Suggest, term=RunWith(U,friend));**

**(communicative\_act name=Remind, term=V-Goal(Giuseppe GoodHealth));**

*S4: Grazie, sei gentile. Chissà, in un mondo virtuale! Per ora, puoi andare a correre con un'amica, se vuoi restare in buona salute...*

*U4: Ma ti pare che a vent'anni devo già pensare alla salute?*

Input Interpretation interpreta la User move come rifiuto della motivazione.

S capisce che la sua strategia persuasiva è fallita e cerca di riparare cercandone una alternativa. Trova che un'altra ragione per correre è l'essere in forma. Aggiunge all'agenda un nuovo piano di persuasione:

**(communicative\_act name=Inform, term=Implies(Run InShape))**

*S5: Ma correre tiene anche in forma!*

59

## Esercizio 3

Sviluppa l'esempio precedente piu' in dettaglio, e cioè definendo:  
update-rule, select-rule, rappresentazione dello user model, storia del dialogo ecc.

60

## Esercizio 4

Sviluppa nei dettagli questo esempio:

S1: Mangi la frutta Maria?

U1: Sì, ne vado matta!

S2: Hai ragione, soprattutto in estate la frutta è fantastica!  
E i vegetali, li mangi?

...

Nota che il piano iniziale di S è, in questo caso, raccogliere informazioni sulle abitudini alimentari di U (se mangia frutta e vegetali).

Il piano viene messo momentaneamente da parte da S per rispondere con gentilezza a quello che interpreta una manifestazione di 'social attitude' da parte di U

61

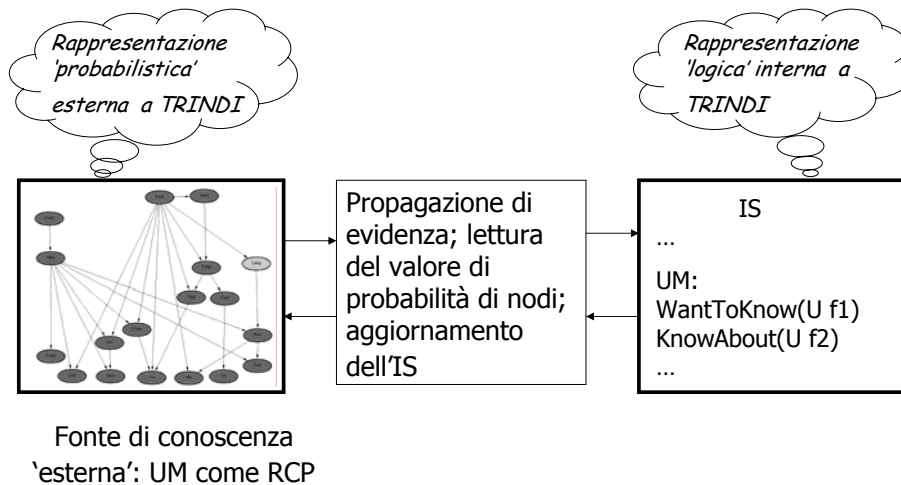
## Collegamento con moduli esterni

Possono essere diversi, a seconda del tipo di dialogo che si vuole simulare:

- un generatore di linguaggio naturale più sofisticato
- un player di agenti animati conversazionali
- un modulo di user modeling basato su RCP
- un modulo di selezione di strategie persuasive
- un parser delle move dell'utente
- ...

62

Un esempio: come realizzare la consistenza fra modalità diverse per la rappresentazione della conoscenza relativa all'utente



63

## Riferimenti

- D R Traum e S Larsson: The information state approach to dialogue management  
<http://www.di.uniba.it/intint/people/ArticoliDid/Dialogo.zip>

e, per chi ha un interesse particolare sull'argomento, la Tesi di PhD di Timothy Bickmore (all'MIT-Boston), scaricabile dalla rete.

64