

Laurea Specialistica in Informatica  
a.a. 2007-2008

Interazione Uomo-Macchina II:

Interfacce Intelligenti

Fiorella de Rosis  
Berardina De Carolis

Laboratori di  
Irene Mazzotta e Nicole Novielli

1

Introduzione

Unità 1: Ragionamento logico:  
- Formalizzazione  
- Risoluzione

Unità 2: Generazione di linguaggio naturale

- Teorie  
- **Metodi**

Unità 3: Ragionamento incerto

- Reti Causali Probabilistiche  
- Reti dinamiche

Unità 4: Modelli di Utente

- Modelli logici  
- Modelli con incertezza

Unità 5: Comprensione del linguaggio naturale

Unità 6: Simulazione di dialoghi

- Modelli basati su ATN  
- Modelli basati su IS

Unità 7: Affective Computing

Programma  
del Corso

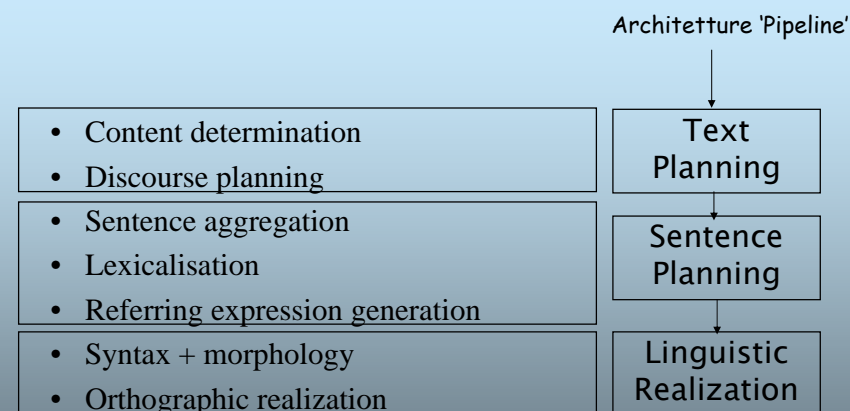
2

Di cosa parleremo in questa Unità

- Architetture di NLG
- di pianificazione del discorso
  - Pianificazione per regressione dal goal
  - Pianificazione per livelli di astrazione
- Piani come strutture XML
- Aggregazione di ripetizioni
- Realizzazione superficiale
- Integrazione con il player di un Agente Animato
- Generazione di singoli atti comunicativi con RealPro

3

Task e Architetture NLG



4

## 1. Text Planning

### \* Content Determination

Il processo che porta a stabilire *cosa dire nel messaggio*

### \* Discourse Planning

Il processo che porta a stabilire

- in quale ordine le informazioni che compongono il messaggio devono essere presentate e
- quali sono le relazioni fra queste informazioni

5

## Il Piano del Discorso riflette la struttura intenzionale e retorica di un testo: un primo esempio

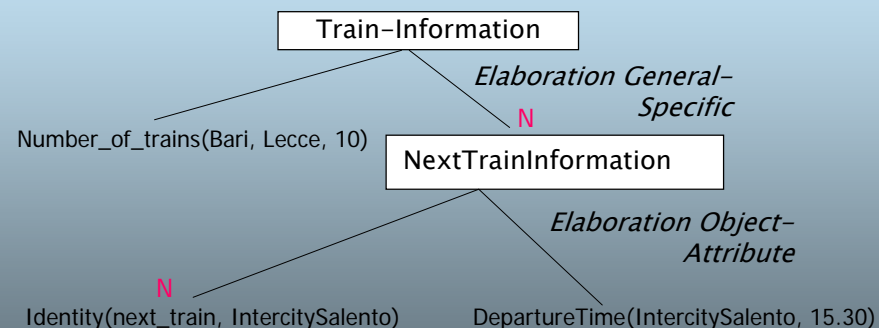
DS0

DS1 Ci sono 10 treni che viaggiano quotidianamente da Bari a Lecce.

DS2

DS2.1 Il prossimo treno è l'Intercity "Salento".

DS2.2 Parte da Bari Stazione Centrale alle 15.30.



6

## 2 Sentence Planning

### \* Sentence aggregation

Un mapping 1:1 dai messaggi alle frasi può generare un testo ridondante e 'artificioso'. I messaggi devono essere aggregati in modo da migliorare la qualità del testo generato

Esempio:

- Senza aggregazione:
  - Il prossimo treno è l'Intercity "Salento". L'intercity "Salento" parte alle 15.30 da Bari.
- Con aggregazione:
  - Il prossimo treno è l'Intercity "Salento" che parte alle 15.30 da Bari.

### \* Lexicalisation:

Vengono definite le parole da utilizzare per esprimere concetti e relazioni in un particolare dominio.

- Esempio:
    - DepartureTime: 'parte alle'
    - next\_train: 'il prossimo treno'
- qui s'inserisce la scelta della lingua

## 3 Linguistic Realisation

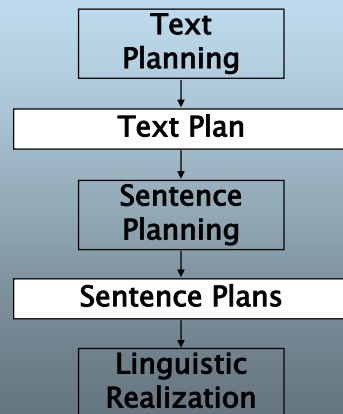
*Si stabilisce:*

- con quale *'mezzo'* rendere ogni elemento (linguaggio naturale, immagine, voce, ...)
- quale *stile* usare (colloquiale, formale, ...)
- quali *frasi* usare
- quali *immagini*
- in che *posizione relativa* mettere i diversi elementi
- quale *formato* dare ad ogni parte del documento (caratteri, dimensioni, formattazione,...)

8

## Noi distingueremo (semplificando) due task principali

- *definizione del 'piano del discorso'*: cosa dire e in che ordine dirlo;
- *generazione 'superficiale'*: come dirlo.



9

## Il metodo 'classico per costruire il Piano del discorso Pianificazione per 'Regressione dal Goal'

E' una procedura di pianificazione che utilizza

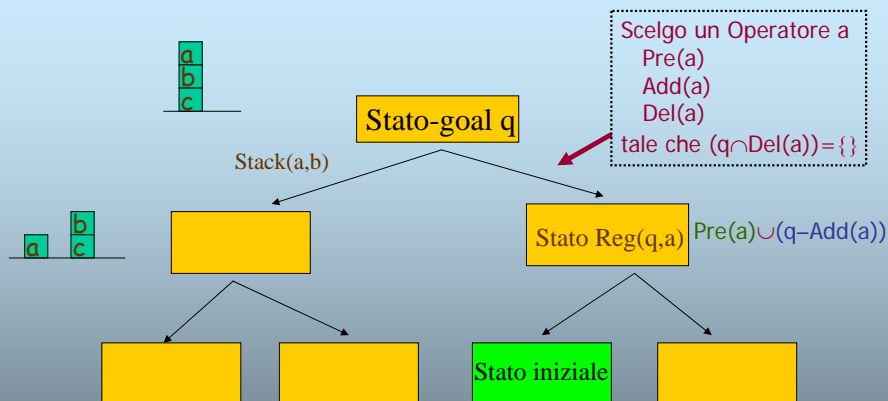
- un metodo 'backward' di ricerca della soluzione, e cioè a partire dal goal, fino a raggiungere lo stato iniziale,
- Un particolare formalismo per rappresentare gli operatori e gli stati

*Operatore A:*

- Pre(A): lista di condizioni che devono essere verificate perché l'operatore sia applicabile;
- Add(A): lista di proprietà che diventano vere dopo l'applicazione dell'operatore
- Del(A): lista di proprietà che diventano false dopo l'applicazione dell'operatore

10

## L'esempio più classico: il mondo dei blocchi



11

## Metodo di ricerca della soluzione nella regressione dal goal

**Criterio:**

- applicare operatori che evitino di cancellare *proprietà da raggiungere*,
- descrivere il nuovo stato-goal mediante l'insieme delle *precondizioni dell'operatore* e delle *proprietà non ancora raggiunte*.

A partire dallo stato goal,

si applica ricorsivamente al generico stato q una procedura che:

- cerca un operatore a applicabile  
(come l'operatore la cui parte Del ha una intersezione vuota con la descrizione di q),
- determina il nuovo stato Reg(q,a):  
(come lo stato descritto dall'unione di Pre(a) e q-Add(a))

$$(q \cap \text{Del}(a)) = \{\} \Rightarrow \text{Reg}(q,a) = \text{Pre}(a) \cup (q - \text{Add}(a))_{12}$$

12

## Pianificazione di discorsi

Un atto comunicativo può essere visto come un'azione che uno Speaker (S) compie nei confronti di un ascoltatore (H) con l'*obiettivo di cambiare il suo stato mentale*.

Esempio:

- con una *'inform'* su un determinato fatto, S si pone l'obiettivo di far sì che H creda in quel fatto,
- con una *'answer'* su un determinato fatto, S si pone lo stesso obiettivo,
- con una *'suggest'* su una azione, si pone l'obiettivo che H compia quella azione, ecc.

Ma dietro ogni azione c'è anche una *ipotesi di S sullo stato mentale di H*:

- dietro una *'inform'*, c'è l'ipotesi che H non creda già in quel fatto,
- dietro una *'answer'*, c'è l'ipotesi che H desideri conoscere il valore di verità del fatto e che non lo conosca già (ad es perché H gliel'ha chiesto),
- dietro una *'suggest'* c'è l'ipotesi che H non abbia già l'intenzione di compiere quell'azione, ecc.

Pianificare un discorso, quindi, non è molto diverso dal pianificare una qualsiasi attività complessa.

13

## Un modo di vedere il problema della pianificazione di discorsi

**Descrizione degli stati:**

Combinazione logica di elementi dello stato mentale dell'utente

(lo vedremo meglio nell'Unità sui modelli di utente):

predicati: KnowAbout, WantToKnow, Prefer, Like, IsInterestedIn, KnowHow, CanDo, oppure belief

**Descrizione degli operatori:**

Atti comunicativi:

Inform(S,U,f): informare su un fatto f

Persuade(S,U,a): persuadere a compiere una azione a

Instruct(S,U,a): istruire su come compiere una azione a

...

descritti come operatori (Pre & Add & Del)

*Il Piano del Discorso consiste (nella sua formulazione più semplice in una concatenazione di atti comunicativi che permette di realizzare lo stato-goal a partire dallo stato-iniziale*

## Pianificazione di discorsi: un esempio

**Descrizione dello stato-goal:**

*l'utente deve avere l'intenzione di compiere l'azione A e deve sapere come farlo*

WantToDo(U,A)  
KnowHow(U,A)

**Descrizione dello stato iniziale:**

¬WantToDo(U,A)  
¬KnowHow(U,A) *l'utente non intende compiere A e non sa come compierla*

**Descrizione degli operatori:**

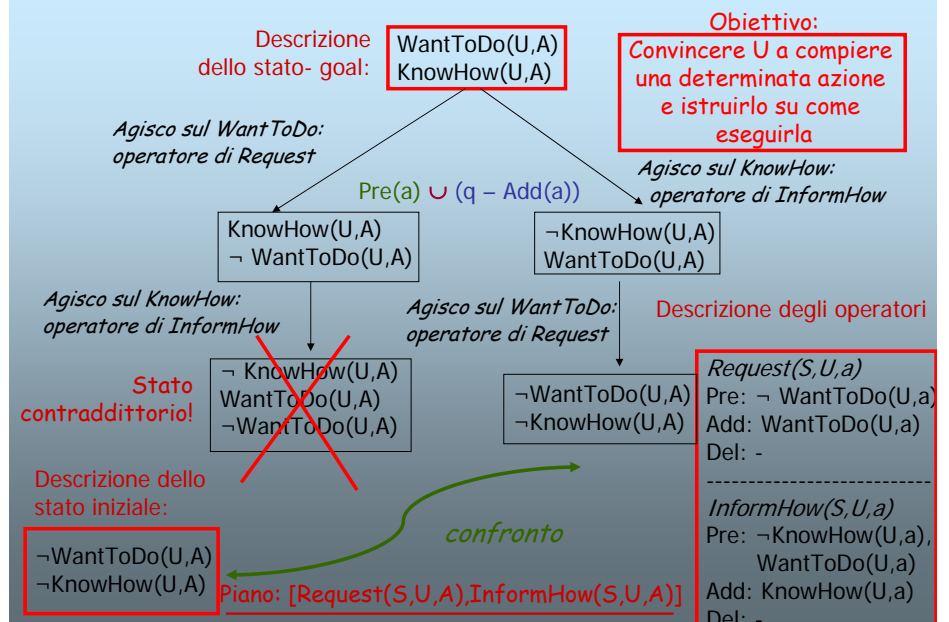
Request(S,U,a)  
Pre: ¬ WantToDo(U,a)  
Add: WantToDo(U,a)  
Del: -

=====

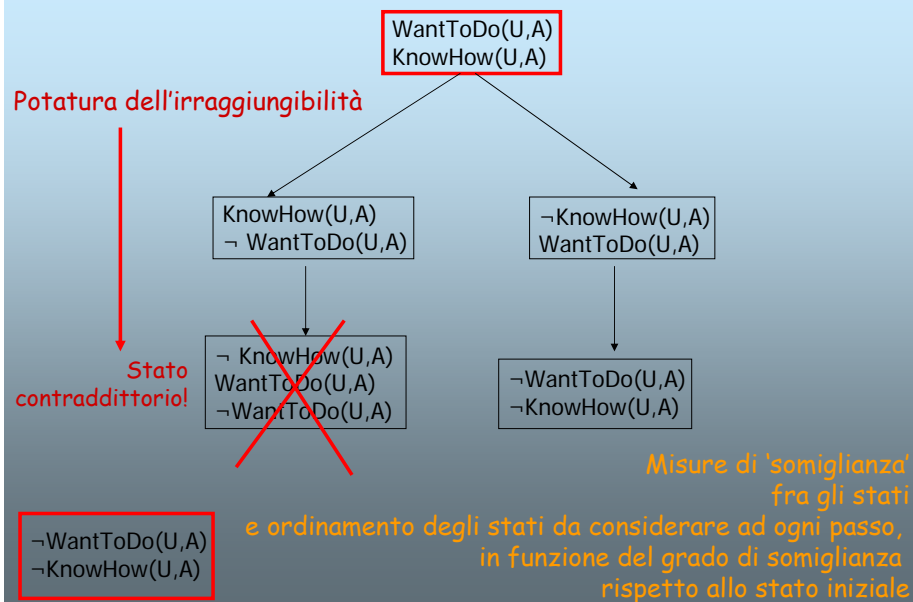
InformHow(S,U,a)  
Pre: ¬KnowHow(U,a),  
WantToDo(U,a)  
Add: KnowHow(U,a)  
Del: -

15

## Pianificazione di discorsi per regressione dal goal



## Metodi per la riduzione dello spazio di ricerca:

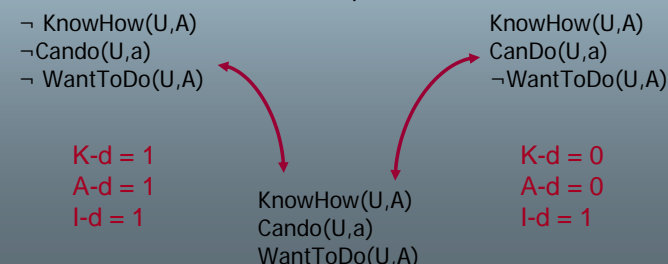


## Misure di somiglianza nel dominio del discorso

Consistono nel 'misurare' la distanza fra stati in termini di differenza fra gli elementi dello stato mentale dell'utente:

Knowledge-difference  
Intention-difference  
Ability-difference  
...ecc

Esempio:



18

## Quali problemi nel metodo di regressione dal goal?

- Lo spazio di ricerca può essere molto complesso, e quindi la ricerca della soluzione può richiedere uno spazio e un tempo inaccettabili, se il problema non è banale (goal-state 'ricco' e molti atti comunicativi disponibili).
- Ci possono essere molti modi per ottenere uno stesso effetto: ad esempio, una Request si può formulare in termini di Order, di Suggest, di Praise,...

*Soluzione:*

applicare un metodo di pianificazione diverso, basato su una *ricerca di soluzioni 'per livelli di astrazione'*.

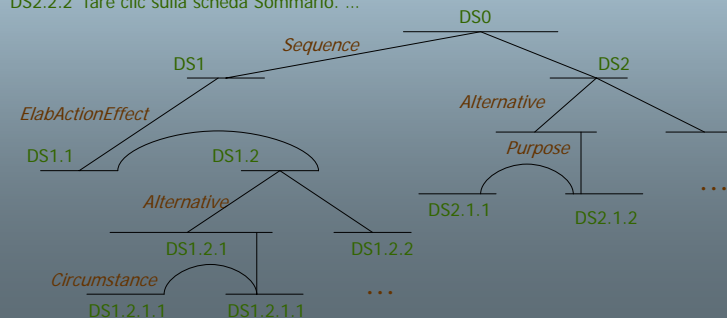
**Cioè, decomporre il problema in sottoproblemi fino ad individuare problemi semplici, affrontabili con un solo operatore**

Riprendiamo qualcuno dei nostri esempi

19

## La struttura RST del messaggio di help

DS0  
DS1  
DS1.1 Scegliere Guida in linea Microsoft Word dal menu ?.  
DS1.2  
DS1.2.1 Se è abilitato, verrà visualizzato l'Assistente.  
DS1.2.2 Se l'Assistente è disabilitato, verrà visualizzata la finestra della Guida.  
DS2  
DS2.1  
DS2.1.1 Per digitare una domanda nella finestra della Guida,  
DS2.1.2 fare clic sulla scheda Ricerca libera.  
DS2.2  
DS2.2.1 Per sfogliare il sommario della Guida,  
DS2.2.2 fare clic sulla scheda Sommario. ...

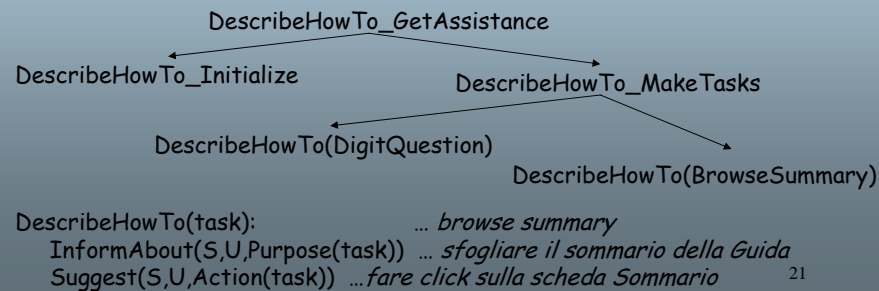


20

## Rappresentiamola come Piano

Scegliere Guida in linea Microsoft Word dal menu ?.  
 Se è abilitato, verrà visualizzato l'Assistente.  
 Se l'Assistente è disabilitato, verrà visualizzata la finestra della Guida.  
 Per digitare una domanda nella finestra della Guida,  
 fare clic sulla scheda Ricerca libera.  
 Per sfogliare il sommario della Guida,  
 fare clic sulla scheda Sommario. ...

### Decomposizione di goal in sub-goal:



21

Si formulano gli stati in modo 'astratto' anziché per descrizione dello stato mentale dell'utente:

DescribeHowTo(GetAssistance) si decompone in:

DescribeHowTo(Initialize) e  
 DescribeHowTo(MakeTasks)

DescribeHowTo(MakeTasks) si decompone in:

DescribeHowTo(DigitQuestion)  
 DescribeHowTo(BrowseSummary)

fino a decomporre in 'azioni elementari' che corrispondono a singoli 'atti comunicativi':

InformAbout(S,U,x), con x=formula  
 condition:  $\neg \text{KnowAbout}(U,x)$   
 effect:  $\text{KnowAbout}(U,x)$

Suggest(S,U,a), con a=termine  
 condition:  $\neg \text{WantToDo}(U,x)$   
 effect:  $\text{WantToDo}(U,x)$

22

## La struttura RST dell' esempio persuasivo

DS0

DS1 Migliore postura

DS2

DS3

DS3.1 Una postura sbagliata è la causa principale di infortuni o inconvenienti vari

DS3.2

DS3.2.1 Spesso è anche la causa dell'insorgere di problemi più gravi

DS3.2.2 con l'avanzare dell'età

DS4

DS4.1

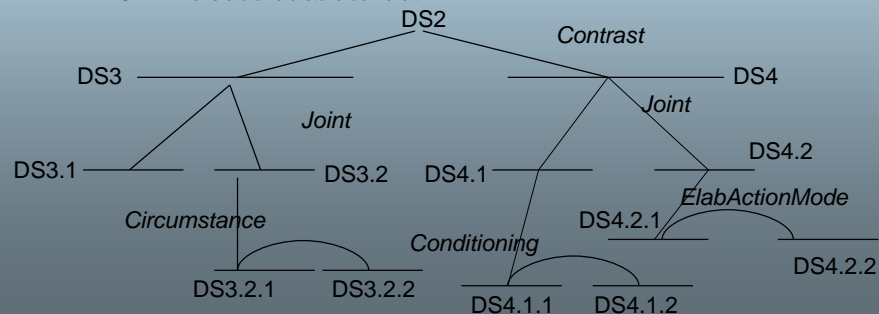
DS4.1.1 Fare esercizi favorisce una postura migliore

DS4.1.2 (se questi esercizi sono eseguiti) nella forma corretta

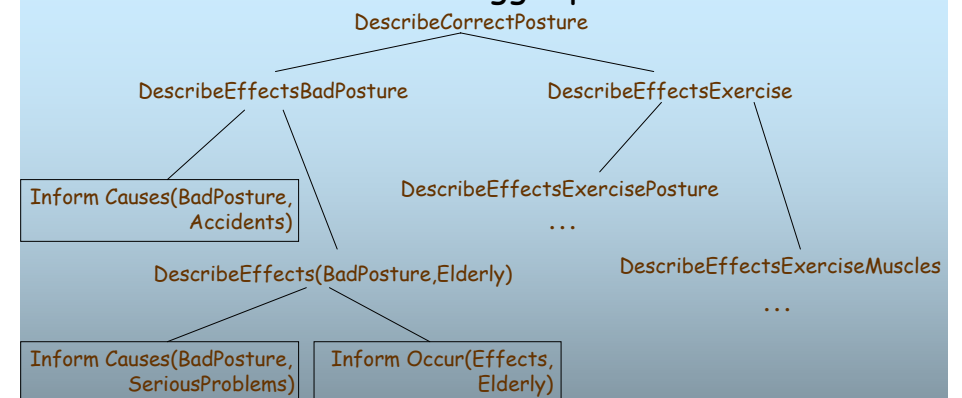
DS4.2

DS4.2.1 assieme al rafforzamento e all'allungamento dei muscoli

DS4.2.2 che aiutano a stare ben eretti



## Piano del messaggio persuasivo



24

## Relazione con la Teoria

Cosa ritroviamo in questo metodo di pianificazione, delle teorie che abbiamo visto nell'Unità precedente?

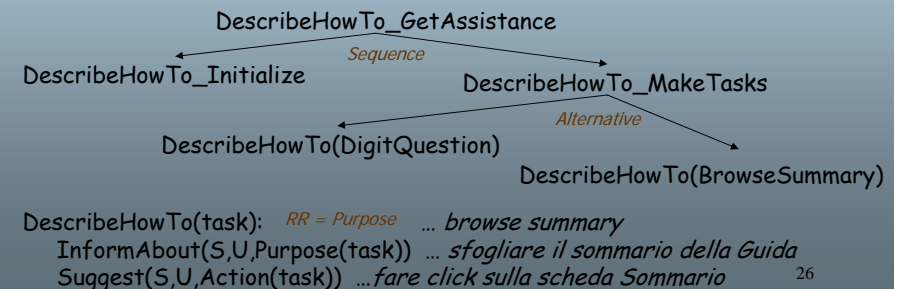
- La *struttura linguistica* di un piano è data dall'insieme delle sue foglie (le frasi che lo costituiscono);
- La sua *struttura intenzionale* è rispecchiata dalla struttura di goal e sub-goal, che indicano con chiarezza le relazioni di precedenza fra i discourse segment DS;
- La sua *struttura attenzionale* (focus dei diversi discourse segment) è rispecchiata dai termini dei goal/subgoal o degli atti comunicativi elementari. Nell'esempio precedente: GetAssistance, MakeTask, DigitQuestion, ecc;
- I *singoli* speech act sono descritti in termini di precondizioni ed effetti.

Notare: se confrontiamo questa rappresentazione con la struttura RST che abbiamo visto nell'Unità precedente, ci accorgiamo che abbiamo perso (per ora) l'informazione sulle RR che legano le varie parti del discorso.

## Inseriamo le RR nel piano del messaggio di help

Scegliere Guida in linea Microsoft Word dal menu ?.  
 Se è abilitato, verrà visualizzato l'Assistente.  
 Se l'Assistente è disabilitato, verrà visualizzata la finestra della Guida.  
 Per digitare una domanda nella finestra della Guida,  
 fare clic sulla scheda Ricerca libera.  
 Per sfogliare il sommario della Guida,  
 fare clic sulla scheda Sommario. ...

### Decomposizione di goal in sub-goal:



## Operatori per la Pianificazione del Discorso

Sono basati sull'idea di rappresentazione di azioni come 'frame'.

**Name:** *un identificatore dell'operatore*  
**Header:** *la descrizione del suo obiettivo*  
**Constraints:** *vincoli (su S e su H) che devono essere soddisfatti perché l'operatore sia applicabile*  
**Effects:** *effetti dell'operatore sulla mente di H*  
**Decomposition:** *la descrizione dei sotto-obiettivi in cui l'obiettivo descritto nello Header può essere decomposto*

Il contenuto dei vari slot è una wff.

## Generazione di Testi: Operatori di Moore e Paris

Ogni operatore contiene i seguenti attributi:

- **Effetto:** obiettivo dell'azione descritta dall'operatore
- **Una lista di vincoli (precondizioni):** condizioni che devono essere vere per poter applicare l'operatore e „garantire“ che produca l'effetto descritto. Queste precondizioni possono riferirsi a fatti nella base di conoscenza sul dominio, nello user model, nella storia del dialogo oppure sullo stato del processo di pianificazione.
- **Un nucleo:** rappresenta il sottogoal più „importante“ in cui si decompone il goal. E' obbligatorio
- **Uno o più satelliti:** rappresentano sottogoal aggiuntivi che contribuiscono a raggiungere l'effetto dell'operatore. Possono essere opzionali.



### Esempio: operatore di piano per la MOTIVATION

- EFFECT: (MOTIVATION ?act ?goal)
- CONSTRAINTS: (AND (STEP ?act ?goal)
- (GOAL ?hearer ?goal))
- NUCLEUS: (BEL ?hearer (STEP ?act ?goal))
- SATELLITES: NIL

### Il pianificatore:

- utilizza un modello degli stati mentali e dei goal comunicativi:
  - (KNOW ?agent (ref ?description))
  - (BEL ?agent (?predicate ?e1 ?e2))
- funziona per espansione gerarchica top-down.

29

## Generazione di Messaggi Multimodali (M.Maybury, MITRE Corporation)

Gli atti comunicativi linguistici vengono estesi  
con atti 'non verbali':  
ad esempio, atti 'fisici' (del corpo)  
e 'grafici' (disegni o immagini)

PHYSICAL ACTS	LINGUISTIC ACTS	GRAPHICAL ACTS
<i>DEICTIC ACT</i>		
Point, tap, circle	Inform,...	Highlight, blink, circle
Indicate direction	Suggest,...	Indicate direction
ATTENTIONAL ACT	Ask,...	DISPLAY CONTROL ACT
Snap/clap fingers, clap hands		Display-region Zoom (in, out)
BODY LANGUAGE ACT		DEPICT ACT
Facial expressions		Depict image
gestures		Draw (line, arc, circle,...)
Sign language		Animate-action

## Operatori Multimodali di Maybury (1)

NAME: *Identify-location-linguistically*

HEADER: Identify(S,H, e)

CONSTRAINTS: Entity(e)

PRECONDITIONS: Goal S, KnowAbout(H, Location<sup>o</sup>(e))

EFFECTS: KnowAbout(H, Location<sup>o</sup>(e))

DECOMPOSITION: Inform(S,H, Location<sup>o</sup>(e))

Condizioni sullo  
Stato mentale  
Dell'utente

NAME: *Identify-location-linguistically&visually*

HEADER: Identify(S,H, e)

CONSTRAINTS: CartographicEntity(e)

PRECONDITIONS: Visible(e) ∧ Goal S, KnowAbout(H, Location<sup>o</sup>(e))

EFFECTS: KnowAbout(H, Location<sup>o</sup>(e))

DECOMPOSITION: IndicateDeictically(S,H, Location<sup>o</sup>(e)) ∧  
Inform(S,H, Location<sup>o</sup>(e))

31

## Operatori Multimodali di Maybury (2)

NAME: *Make-entity-visible*

HEADER: MakeVisible(e)

CONSTRAINTS: CartographicEntity(e)

PRECONDITIONS: Displayed(e)

EFFECTS: Visible(e)

DECOMPOSITION: DisplayRegion(e)

NAME: *Explain-route-linguistically-and-visually*

HEADER: ExplainRoute(S, H, from-e, to-e)

CONSTRAINTS: CartographicEntity(from-e) ∧ CartographicEntity(to-e)

PRECONDITIONS: Visible(from-e) ∧ (Goal S KnowHow(H, Go(from-e, to-e)))

EFFECTS: KnowHow(H, Go(from-e, to-e)) ∧  
∀segment ∈ path Know(H, Subpath(segment, path))

DECOMPOSITION: ∀segment ∈ path:  
IndicateDeictically(S,H,Source<sup>o</sup>(segment))  
Order(S,H,Go(Source<sup>o</sup>(segment),  
Link<sup>o</sup>(segment),Destination<sup>o</sup>(segment))  
Identify(S,H,to-e)

32



## Gli operatori possono essere raffinati

- Distinguendo fra precondition 'essenziali' e 'desiderabili':  
i primi devono essere veri perché l'operatore sia applicabile  
i secondi rappresentano vincoli 'deboli'
- Distinguendo fra subgoal 'necessari' ed opzionali'
- *E aggiungendo uno slot 'Rhetorical Relation'.*

33

## Operatori Multimodali di Maybury (3)

**NAME:** *extended-description*

**HEADER:** Describe (S,H,e)

**CONSTRAINTS:** Entity (e)

**PRECONDITIONS:**

**ESSENTIAL:** KnowAbout(S,e)  $\wedge$  Want(S, KnowAbout(H,e))

**DESIRABLE:**  $\neg$  KnowAbout(H,e)

**EFFECTS:** KnowAbout(H,e)

**DECOMPOSITION:** Define(S,H,e)

**optional:** Detail(S,H,e)  $\wedge$  Divide(S,H,e)  $\wedge$  Illustrate(S,H,e)  $\wedge$  Give-Analogy(S,H,e)

Il goal Describe(S,H,e) si decompone nei sub-goal Define(S,H,e) e (opzionalmente) Detail(S,H,e), Illustrate(S,H,e) e Give-Analogy(S,H,e).  
Ha, come precondizioni, che S conosca l'entità da descrivere e che voglia che anche H la conosca. Quest'ultimo è l'effetto prodotto dall'operatore.

34

## Outline di un Algoritmo semplificato di Pianificazione

### Pattern-matching Preconditions-Effects e Decomposition-Header

Considera il Goal g

Cerca un operatore Op tale che:

Header(Op) unificabile con g

Constraints e Preconditions su U satisfied

Verifica Preconditions(Op);

$\forall$  Pr: Pr  $\in$  Preconditions(Op)

se Pr non verificata:

cerca un Op' tale che: Effect(Op') unificabile con Pr

aggiungi Op' al piano

Verifica Decomposition(Op)

$\forall$  De: De  $\in$  Decompositions(Op)

se De non verificata:

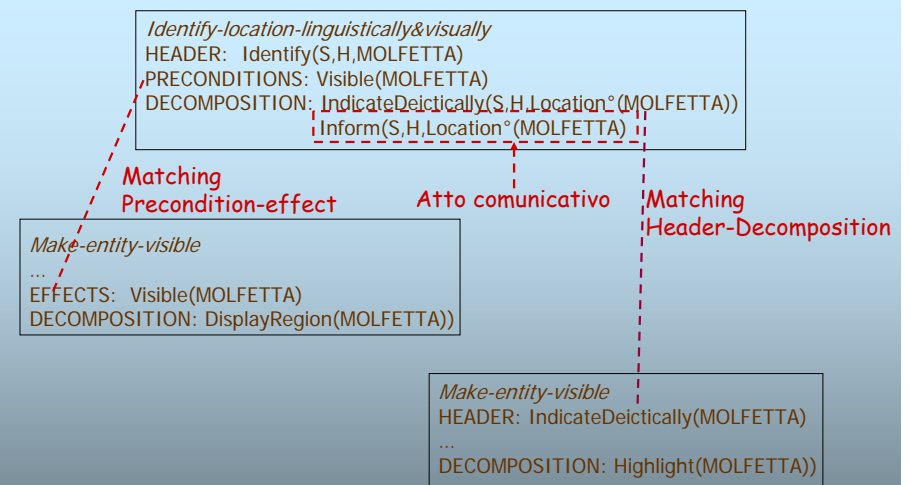
cerca un Op'' tale che: Header(Op'') unificabile con De

aggiungi Op'' al piano

Condizione di stop: De è un *atto comunicativo elementare*.

Notare che la verifica delle condizioni sullo stato mentale dell'utente consente di adattare il piano alle caratteristiche dell'utente

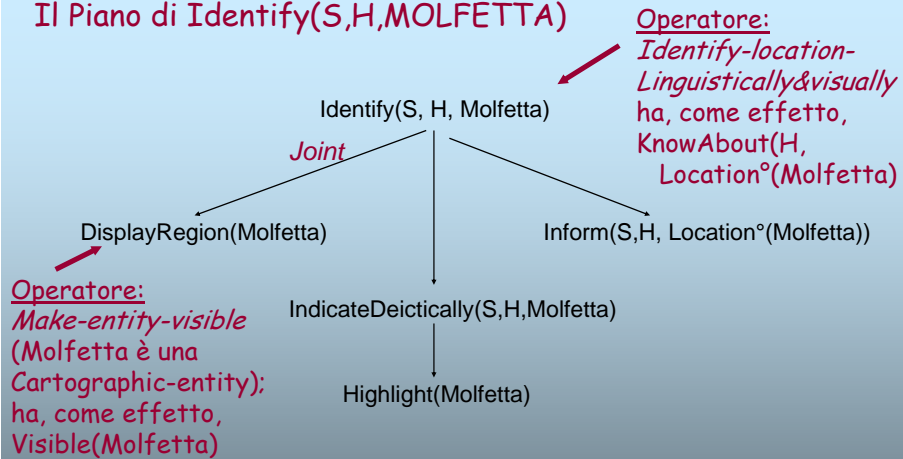
35



Applichiamo il processo di pianificazione al goal:  
Identify(S,H,MOLFETTA)

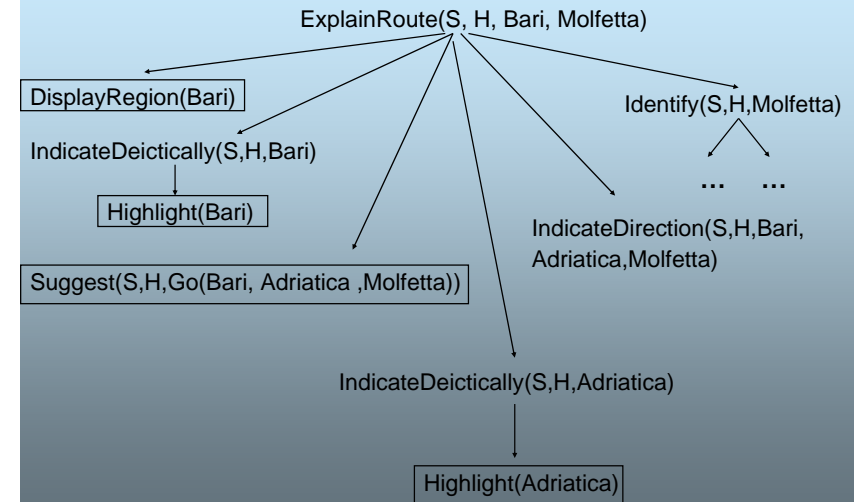
36

## Il Piano di Identify(S,H,MOLFETTA)



37

## Il piano di ExplainRoute(S,H,BARI,MOLFETTA)



38

## La rappresentazione più semplice: Piani del discorso come 'Ricette Preconfezionate'

Quando il messaggio da generare ha una struttura prestabilita, almeno per grandi linee, è possibile rappresentare i piani come 'recipes' non istanziate (strutture ad albero, o anche liste di atti comunicativi, verbali e non verbali), da istanziare di volta in volta con i valori delle variabili che corrispondono al dominio applicativo considerato.

*Esempio:*

Identify(S,H,e):=  
[DisplayRegion(e), Highlight(e), InformAbout(S,H,Location°(e))]

39

## Ma i piani possono essere rappresentati anche come file XML

E' possibile definire un linguaggio xml per la rappresentazione di 'document plans' (DPML)

*Un esempio: il piano generico di Identify(e), con e = 'cartographic-entity':*

```

<xml version="1.0">
<DPML>
<goal name="Identify" term="e" RR="Joint">
  <communicative_act name="DisplayRegion" term="e">
  </communicative_act>
  <goal name="IndicateDeictically" term="e">
    <communicative_act name="Highlight" term="e">
    </communicative_act>
  </goal>
  <communicative_act name="InformAbout" term="Location°(e)">
  </communicative_act>
</goal>
</DPML>
  
```

**L'attributo RR è facoltativo**

A differenza della lista, questa rappresentazione XML rispetta la struttura ad albero del piano e specifica le RR

40

## Aggregation: I piani devono essere 'raffinati'

Un piano conterrà, il più delle volte, *ripetizioni* che renderebbero il discorso pesante e poco naturale.  
Il task di *microplanning* ha, come obiettivo, l'*aggregazione* di parti simili.

Riprendiamo l'esempio delle previsioni del tempo di Repubblica:  
*"Molto nuvoloso o coperto sulla Liguria, basso Piemonte, Emilia Romagna, Bassa Lombardia e basso Veneto, con precipitazioni sparse.... Nuvolosità più attenuata sulle rimanenti zone del Nord,...".*



41

## Vediamo un esempio di aggregazione sul piano

*I ciclamini sono piante da bulbo che fioriscono in inverno.  
Le dalie sono piante da bulbo che fioriscono in estate.  
Le rose sono arbusti che fioriscono in estate.*

Un'aggregazione di parti simili produrrebbe il testo seguente:

*I ciclamini e le dalie sono piante da bulbo, mentre le rose sono arbusti.  
I ciclamini fioriscono in inverno, mentre le rose e le dalie fioriscono in estate.*

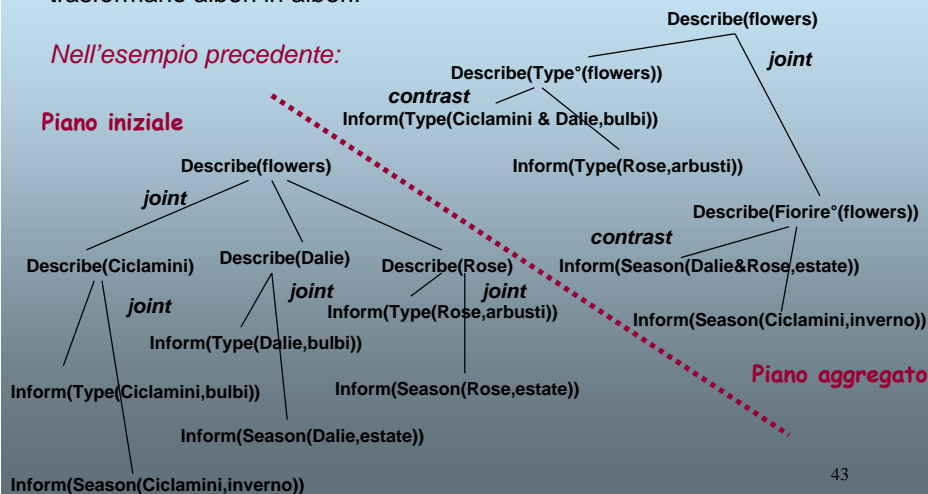
...vediamo come...

42

## Aggregazione come algoritmo su alberi

La funzione di aggregazione, così come altre procedure di microplanning, trasformano alberi in alberi.

Nell'esempio precedente:



43

## Realizzazione superficiale

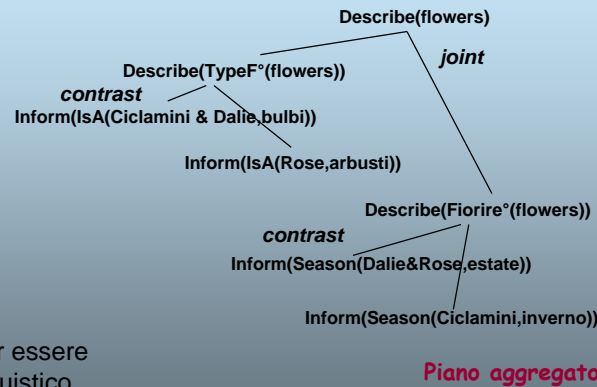
Il piano del discorso è indipendente dalle caratteristiche dell'apparecchiatura sulla quale verrà realizzato, dalla lingua e dalla modalità di output.

La sua traduzione in un messaggio (in linguaggio naturale, ipermediale, multimediale, in forma vocale o mediante un 'agente animato') viene realizzata in due fasi:

- *generazione della struttura*: si stabilisce l'organizzazione del messaggio in paragrafi o 'pagine', capitoli, sezioni;
- *generazione linguistica*: si 'rende' ogni parte del messaggio nella forma voluta, eventualmente arricchendolo con un 'linguaggio di mark-up' (html o altro).

44

## Un semplice esempio di realizzazione multimodale: il piano sui fiori



45

## ... e realizziamolo (in parte) in modo multimediale



46

## Stabiliamo dei criteri generali di generazione della struttura

Rendiamo:

- le *inform* con frasi in linguaggio naturale;
- le *display* con immagini;
- il *focus* del discorso in *italico*;
- le RR di *contrast* con una disposizione spaziale in cui le due descrizioni vengono messe in sequenza;
- le RR di *background* con una disposizione spaziale in cui la frase e l'immagine sono affiancate (come nell'esempio delle previsioni del tempo di Repubblica),
- ecc

47

## ... otterremo il messaggio seguente...

I *ciclamini* e le *dalie* sono piante da bulbo,  
*mentre* le *rose* sono arbusti.

Mentre le *dalie* e le *rose*  
fioriscono in estate,



i *ciclamini* fioriscono  
in inverno.



48

## Altri criteri

Possiamo rendere:

- la *sequence* con una lista ordinata,
- la *elaboration process step* con una lista numerata,
- la *contrast* con due colonne di una table,
- ...

Possiamo enfatizzare il nucleo oppure il satellite delle *motivation* in bold o con una highlight (per dare rilevanza all'azione suggerita o alla ragione per cui la suggeriamo),...

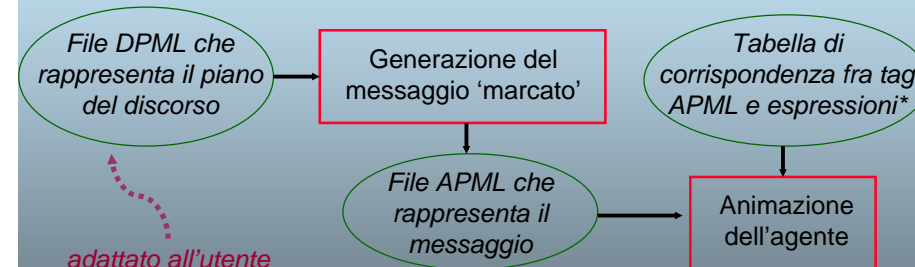
I criteri stabiliti vengono poi tradotti in programmi di generazione del messaggio.

49

## Output come file XML

Uno dei casi in cui conviene produrre, da piani del discorso rappresentati come file xml, nuovi file xml con annotazione diversa, è quello della *animazione di agenti conversazionali*.

In questo caso, lo schema di generazione è il seguente:



\* "meaning-signal"

50

## Un messaggio persuasivo: struttura del piano



## b. rappresentazione DPML

```

<xml version="1.0">
<DPML>
<goal name="Persuade" term="EatFibre" RR="Motivation">
  <goal name="InduceIntention" term="EatFibre" RR="Conditioning">
    <communicative_act name="Inform" term="Desire(U,Health)">
    </communicative_act>
    <communicative_act name="Suggest" term="Eat(U Fibre)">
    </communicative_act>
  </goal>
  <goal name="Describe" term="PositiveEffects°(EatFibre) RR="Joint">
    <communicative_act name="InformAbout" term="Effects°(Fibre, wellbeing)">
    </communicative_act>
    <communicative_act name="InformAbout" term="Effects°(Fibre, tooth)">
    </communicative_act>
  </goal>
</DPML>
  
```

Ora vediamo quale input per l'agente animato occorre generare

52

## b. Trasformazione degli atti comunicativi in frasi

```
<xml version="1.0">
<DPML>
<goal name="Persuade" term="EatFibre" RR="Motivation"> Se
  <goal name="InduceIntention" term="EatFibre" RR="Conditioning">
    <communicative_act name="Inform"
      term="Desire(U,Health)"> ci tieni ad avere un cuore sano e un buon
      </communicative_act> apparato digerente
    <communicative_act name="Suggest" term="Eat(U Fibre)">
      </communicative_act> non puoi fare a meno di una giusta dose di fibre
  </goal>
  <goal name="Describe" term="PositiveEffects°(EatFibre) RR="Joint">
    <communicative_act name="InformAbout"
      term="Effects°(Fibre, wellbeing)"> che ti danno un piacevole
      </communicative_act> senso di leggerezza.
    <communicative_act name="InformAbout"
      term="Effects°(Fibre, tooth)"> </communicative_act>
  </goal>
</DPL> e ti aiutano ad avere denti sani e forti.
```

Introduciamo ora nelle frasi i tag necessari per l'animazione dell'ECA

## c. messaggio come file APML

```
<?xml version="1.0"?>
<APML>
  <performative type="inform"> Se ci tieni ad avere <topic-comment
    type="comment">
      un cuore sano e un buon apparato digerente, </topic-comment>
      non puoi fare a meno di una
    <topic-comment type="comment"> giusta dose di fibre,
    <affactive type="happy-for"> che ti danno un piacevole senso di
      leggerezza</affactive>
    </topic-comment>
    Le fibre, inoltre,
    <topic-comment type="comment"> ti aiutano ad avere denti sani e
      forti.</topic-comment>
  </performative>
</APML>
```

Abbiamo introdotto:  
 Un tag che indica l'atto comunicativo ('performative')  
 Un tag che indica l'emozione ('affactive')  
 Un tag che indica l'enfasi ('topic-comment').  
 Questi tag verranno letti dal graphic player e dal TTS e  
 tradotti in espressioni del viso e intonazioni della voce  
 dell'ECA

## d. Tabella di corrispondenza fra tag APML ed espressioni

```
<?xml version="1.0" ?>
<Tagging>
<setup>
<agent name="sally.haptar" />
<volume start="80" /> <speed start="0" /> <voice
  type="female"/>
</setup>
<performative type="inform"> alla inform non associamo
  <animation /> nessuna particolare animazione </performative>
<performative type="suggest"> - <animation>
  <switchON>M_HeadAside</switchON> alla suggest associamo una
  <switchON>M_Frown</switchON> inclinazione laterale del
  </animation> <TTSapi5 /> capo e un aggrottare delle
  </performative> sopracciglia
<performative>
<affactive type="happy-for"> - alla 'happy-for' associamo
  <animation> <switchON>M_Smile</switchON> un sorriso
  </animation> <TTSapi5 /> </affactive> </Tagging>
```

Inserire collegamento a Valentina

## Realizzazione linguistica

Il metodo più semplice consiste nell'associare ad ogni atto linguistico una frase preconfezionata, come abbiamo fatto finora.

Inform(IsA(Ciclamini & Dalie,bulbi)) → "I ciclamini sono piante da bulbo".

In questo caso, però, dovrò memorizzare tante frasi quanti sono gli atti comunicativi da realizzare:

Inform(IsA(Rose,arbusti)) → "Le rose sono arbusti"

Anzi, se intendo realizzare messaggi in lingue diverse, dovrò memorizzare una frase per ogni lingua.

Un'alternativa conveniente è quella di definire un algoritmo per la traduzione dell'argomento dell'atto linguistico in (un frammento di) frase in linguaggio naturale:

IsA(Ciclamini & Dalie, bulbi)	IsA(Rose,arbusti))
Predicato → verbo: 'sono'	
1° termine → soggetto: 'i ciclamini e le dalie'	'le rose'
2° termine → complemento: 'piante da bulbo'	'arbusti'



## Realizzazione linguistica (continua)

Flourish (Dalie&Rose,estate):

**Predicato** → verbo: *'fioriscono'*

1° termine → soggetto: *'le dalie e le rose'*

2° termine → complemento: *in estate*

Con questo metodo:

Data una tabella che associa

ad ogni predicato e ad ogni termine un 'frammento di frase', si possono costruire frasi più complesse.

**Esempio:**

**Suggest(S,H,Go(Bari, Molfetta, Adriatica))**

*"Per andare da Bari a Molfetta, ti suggerisco di prendere l'Adriatica."*

Ma la realizzazione di frasi più complesse richiede un generatore molto più raffinato: ad es RealPro

57

## Rappresentazione dell'input in RealPro:

Una rappresentazione sintattica di frasi semplici o complesse, in una struttura ad albero con archi labellati e nodi labellati con 'lessemi'. Tutti i nodi sono 'terminali'.

La struttura dell'input:

DSYNTS:

```
'root node label'
  ('dependency arc'
   'node'
  )
)
```

END

Con: 'dependency\_arc' = [list of feature:value pairs];  
idem per 'node'

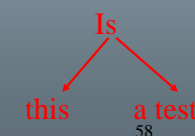
Esempio:

DSYNTS:

```
BE [ ]
  (I THIS [class:demonstrative_pronoun number:sg]
   II test [class:common_noun article:indef]:
  )
)
```

END

Sta per "This is a test"



## I lessemi sono:

### Open-class words

Value of class:	Example
adjective	<i>small, disastrous</i>
adverb	<i>really, maybe, surprisingly</i>
common_noun	<i>table, map</i>
proper_noun	<i>John, Poona, Socks</i>
verb	<i>disintegrate, indulge</i>
symbol	+

Number: s/pl  
Gender: m/f  
Case: declinazione  
Determiner type: quantifier

tense: pres/past/future  
voice: act/pass  
aspect: simple/cont  
mood: ind/cond/imp/pres-part/  
past-part/...  
polarity: nil/neg  
question.: -/+  
...

### Closed-class words:

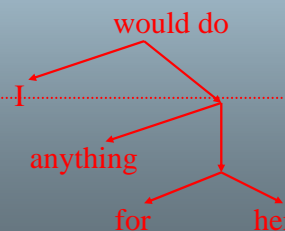
Article  
Demonstrative\_pronoun  
Numeral  
Indefinite\_pronoun  
Preposition  
Quantifier  
....

59

## Esempi

```
DSYNTS:
DO [mood:cond]
(
  I "<PRONOUN>" [number:sg person:1st]
  II anything [class:indefinite_pronoun person:3rd number:sg]
  (
    ATTR FOR [ ]
    (
      II girl [class:common_noun gender:fem article:def pro:pro]
    )
  )
)
)
END:
```

"I would do anything for her"



60

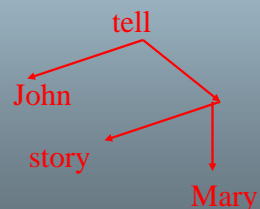


```

DSYNTS:
tell [class:verb]
(
  I John [class:proper_noun]
  III Mary [class:proper_noun]
  II story [class:common_noun]
)
END:

```

"John tells Mary a story"



61

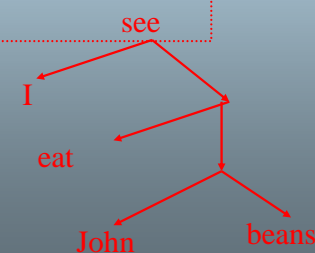
## Embedded clauses

```

DSYNTS:
see [class:verb tense:past morpheme:saw inflection:inv]
(
  I I [class:personal_pronoun person:1st number:sg]
  II eat [class:verb mood:pres-part]
  (
    I John [class:proper_noun]
    II been [class:common_noun number:pl]
  )
)
END:

```

"I saw John eating beans"



62

## Una demo in linea

### RealPro Demo: Variations on "John loves Mary"

Generator Options	Tense	Progressive	Perfect	Question	Negation	Focus on Mary
	<input checked="" type="radio"/> Present <input type="radio"/> Past <input type="radio"/> Future	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes
Output	John loves Mary.					
Generator Options	Tense	Progressive	Perfect	Question	Negation	Focus on Mary
	<input checked="" type="radio"/> Present <input type="radio"/> Past <input type="radio"/> Future	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes
Output	John is loving Mary.					
Generator Options	Tense	Progressive	Perfect	Question	Negation	Focus on Mary
	<input checked="" type="radio"/> Present <input type="radio"/> Past <input type="radio"/> Future	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes
Output	John is not loving Mary.					
Generator Options	Tense	Progressive	Perfect	Question	Negation	Focus on Mary
	<input type="radio"/> Present <input type="radio"/> Past <input checked="" type="radio"/> Future	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes	<input type="radio"/> No <input checked="" type="radio"/> Yes	<input type="radio"/> No <input type="radio"/> Yes
Output	John will not love Mary.					

### Input in RealPro

```

// -----
// John loves Mary.
// -----
DSYNTS:
love [ class:verb ]
( I John [ class:proper_noun ]
  II Mary [ class:proper_noun ]
)
END:

// -----
// John does not love Mary.
// -----
DSYNTS:
love [ class:verb tense:pres polarity:neg ]
( I John [ class:proper_noun ]
  II Mary [ class:proper_noun ]
)
END:

```

63

## Possiamo rendere atti comunicativi diversi (in RealPro) agendo sulle feature dei verbi

```

tense: pres/past/future
voice: act/pass
aspect: simple/cont
mood: ind/cond/imp/pres-part/
      past-part/...
polarity: nil/neg
question.: -/+
...

```

Inform: *'Le rose sono arbusti'*

Mood = ind

AskIf: *'Quanto pesi?'*

Question = +

Request: *'Dovresti andare a correre'*

Mood = imp / cond

64

## Outline di un algoritmo di generazione

Esploro l'albero di piano in modo breadth-first

Per ogni foglia: chiamo RealPro

Per ogni nodo intermedio: faccio un table look-up per inserire il frammento linguistico che corrisponde alla RR associata

## Riferimenti

Building natural language Generation Systems.  
E.Reiter e R.Dale.  
Cambridge University Press, 2000.

### Articoli reperibili sul sito:

G Wilcock: Pipelines, templates and transformations: XML for natural language generation.  
<http://www.di.uniba.it/people/ArticoliDid/XML-NLG.zip>

B.Lavoie and O.Rambow: A fast and portable realizer for text generation systems.  
<http://www.di.uniba.it/people/ArticoliDid/realpro.pdf>

Una descrizione completa (con una piccola demo) di RealPro è visibile su:  
<http://www.cogentex.com/technology/realpro/index.shtml>